DEVELOPMENT OF RELATIVISTIC ELECTRONIC STRUCTURE METHODS FOR ACCURATE CALCULATIONS OF MOLECULES CONTAINING HEAVY ELEMENTS

by
Ayush Asthana

A dissertation submitted to Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
July 2021

Abstract

The dissertation focuses on an efficient implementation of relativistic spin-orbit coupled-cluster methods (SO-CC) widely applicable to molecules containing heavy elements. SO-CC methods have high computational time and storage requirements with a bottleneck associated with the storage and processing of large molecular orbital (MO) integral matrices. These high computational requirements limit the application of SO-CC methods to relatively small molecules compared with their non-relativistic counterparts. Inspired by atomic orbital (AO)-based algorithms in non-relativistic methods, AO-based algorithms have been developed to enhance the computational efficiency of SO-CC methods in the framework of the exact two-component (X2C) theory, with the following advances:

1. The AO-based scheme avoids the evaluation and storage of large MO integral matrices.

2. It lowers the formal floating-point operation count of the computationally significant "ladder term" by a factor of four.

3. It allows the use of sparsity in the AO integral matrix to further reduce the storage requirements and formal operation count.

This dissertation develops the formulation and implementation of the AO-based algorithms for SO-CC methods, leveraging the spin-free nature of AO two-electron integrals and sparsity in the AO integral matrix to eliminate the storage bottleneck and reduce the formal operation count. This implementation has been parallelized using shared memory (OpenMP)-based parallelization.

In addition, the development of an automatic expression generation library, named AutoGen, and its application to the derivation of working equations in unitary coupled-cluster (UCC) singles and doubles-based third-order polarization propagator theory (UCC3) is discussed in the dissertation. Derivation and implementation of working equations has become a limiting factor in developing several classes of quantum chemistry methods. The number of tensor contraction expressions reaches hundreds and even thousands in many methods including the UCC-based methods. Derivation and implementation of such a large number of expressions is time-consuming and error-prone. The Python-based library developed is driven by string-based manipulation of creation and annihilation operators to bring them to normal order using Wick's theorem. Working equations can be extracted in a simple object form, allowing easy extension and integration with other software packages.

Thesis Advisor

Dr. Lan Cheng
Assistant Professor
Department of Chemistry
Johns Hopkins University

Additional Readers

Dr. Harris J. Silverstone
Professor
Department of Chemistry
Johns Hopkins University

Dr. Art Bragg
Associate Professor
Department of Chemistry
Johns Hopkins University

Dedicated to my parents.

Acknowledgements

The past five years spent at Johns Hopkins University have been transformative and humbling. It is surprising how my graduate school journey has given me skills in research and has taught me lessons to live life in harmony. Something I have realized during this process is - "No success is yours alone; it is also your team's success". I find myself fortunate to share my life with wonderfully kind people who form my "team". This section is dedicated to thanking them for their contributions behind the scene to make my PhD a success.

I am grateful to my advisor, Professor Lan Cheng, for his patient guidance and comprehensive support for the growth of my scientific and academic skills during the last five years. Time spent with him has, directly and indirectly, taught me to think deeply regarding scientific problems and ask important questions to explore further into the problem. I thank my committee members, Professor Harris Silverstone and Professor Art Bragg, for their valuable feedback to improve my research and help me grow as a scientist. My lab member, Dr. Junzi Liu, has been a great collaborator and colleague. I enjoyed discussing and working on interesting scientific problems with him for several years at Cheng lab. I would also like to thank my collaborators, Professor Debashis Mukherjee, Dr. Gaoxiang Liu and Professor Kit Bowen, for their excellent research work and discussions. I enjoyed the company of my labmates Xuechen Zheng, Chaoqun Zhang, and Yue Shen. I thank them for their valuable feedbacks during many of my presentations and research discussions.

Outside of work, I have found constant support and love from friends. Pratik, Shoan, and Dhruv have proved to be like another family to me, who have supported me during every difficult step. I cherish the memory of our trips together to national parks and

beautiful cities and many long discussions on varied topics. Aditya has brought with him his constant calm throughout my stay in Baltimore, along with tough competition in sports and the gym. Saurov has been a great friend and colleague with whom I have had extensive discussions regarding research in theoretical chemistry. My close friends Ankitha, Likitha, Vinay, Ashutosh, Tomojit, Harsh, Ayush, Chandan, Arvind, and many more have made the last five years more enjoyable.

I am fortunate to find great joy in science, perhaps due to early inspiration by my teachers and advisors. I would be doing something very different had it not been for Professor Srihari Keshavamurthy in my undergraduate university. The rigour with which he expressed the beauty of quantum theory in his classes made an early mark. My journey in quantum chemistry started in the lab of Professor T. Daniel Crawford at Virginia Tech in the summer of 2014. In his lab, I recognized the importance of new technology in modern research. This was followed by my stays in the lab of Professor Debashis Mukherjee, who is not only a kind advisor to me but also my inspiration to continue research in quantum chemistry. I also thank Professor Shalabh, who has provided me with valuable counsel regarding my academic decisions.

My parents, sister, and grandparents made the most crucial contribution by giving me a chance to succeed in life. I grew up in a small city in India. Shielding me from financial and social problems, they have provided me with the freedom necessary to carry my intellectual pursuits. I have fond memories of my grandfather, who told stories of his challenging and principled life while growing up. These stories have proved to be more and more meaningful and helpful as I have aged. During the last five years, my father has guided me at every step of my graduate study. I want to end with a special thanks to my girlfriend Sneha, who has supported me and whose critical inputs regarding language and writing style have improved the presentation of my research.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Computational science is nowadays playing an important role in scientific discovery in chemistry and physics. Many quantum chemistry and molecular dynamics program packages have been developed and are widely used to carry out molecular calculations [1–19]. According to a report by National Energy Research Scientific Computing Center (NERSC), about 30-40% of the time on their supercomputing machines was spent on quantum chemistry and molecular dynamics calculations in the year 2015 [20]. Quantum chemistry has achieved significant success in the past several decades and, at the same time, still faces many challenges. The established quantum chemical methods have been successful in providing accurate results for molecules containing light elements around equilibrium geometries [21–23]. The incorporation of relativistic effects for computations of molecules containing heavy elements has seen a lot of success in the past few decades [24–34], while the development of cost-effective treatments of relativistic effects to achieve the same applicability as the corresponding non-relativistic machinery is still an outstanding challenge. Another challenge lies in the improvement of the electron correlation methods for high-accuracy calculations of thermochemistry [35–38] and global potential energy surfaces [39–46]. Further progress in extending the applicability of correlated quantum chemistry methods to larger molecular systems, which bypasses the steep scaling of computational time and storage requirements with respect to the size of the molecules, is also needed [47–52]. The increasing complexity of several classes of new electron correlation methods under development [53–56] renders it time-consuming and

error-prone to manually derive and implement working equations [57–59]. The quantum chemistry community will also benefit from improved interoperability through standardization of practices in quantum chemistry software development, reducing redundancy and increasing reproducibility in codes, and deployment of current codes on modern supercomputers where such implementation does not exist [60, 61]. This dissertation is focused on advancing the treatment of relativistic effects in electron-correlation calculations. Specifically, we present an implementation of relativistic spin-orbit coupled-cluster methods widely applicable to heavy-element-containing molecules. In addition, we present the work on automatic implementation techniques for electronic structure methods.

The first part of this dissertation in chapters 3 to 5 deals with the development of relativistic electronic structure methods. Relativistic effects are defined as everything arising from the finite speed of light (c=137.0359895(61) au) compared with c= ∞. It had been generally accepted until about 1970s that relativity has little importance in chemistry, e.g., the famous statement by Paul Dirac in 1929, one year after publishing the Dirac equation, stated that *"The general theory of quantum mechanics is now almost complete... [Relativistic effects] give rise to difficulties only when high-speed particles are involved and are therefore of no importance in consideration of the atomic and molecular structure and ordinary chemical reactions"* [62] based on the idea that valence electrons pertinent to chemistry travel at low speed even in heavy atoms. This understanding changed when computational studies of elements in the lower rows of the periodic table established the importance of relativistic effects in atoms and molecules in 1970s and 1980s [28–32, 63, 64]. Special relativity has two important effects on atoms and molecules: scalar relativistic and spin-orbit coupling effects. Scalar relativistic effects lead to the contraction of s and p orbitals. This effect originates from the increase in the mass of the electrons when they travel to the vicinity of the nucleus of heavy elements and acquire high speeds. Importantly, all s and p orbitals, including the valence ones, experience this contraction as the electrons in these orbitals penetrate to the core region. The contraction of s and p orbitals also leads to an increased shielding and the resulting expansion

of d and f orbitals. The second important effect of relativity is the spin-orbit splitting of $l > 0$ (p, d, f, etc.) orbitals that arise due to the interaction of electron spin with the magnetic field produced by the relative motion of electrons and nucleus. Both scalar relativistic effects and spin-orbit coupling are prominent for heavy elements. Rigorous quantum-chemical treatment of relativistic effects can be carried out by using a four-component Dirac-Coulomb(-Breit) (DC(B)) Hamiltonian. Four-component calculations provide both positronic and electronic solutions. Since only electronic solutions are of importance to chemistry, two-component "electrons only" methods have been developed [65–78]. The most promising two-component method for molecular applications is the exact two-component (X2C) method [67–69, 74]. Both four and two-component Hamiltonian in occupation number representation take the same form as the non-relativistic Hamiltonian. Therefore, relativistic electronic structure methods can be formulated using existing non-relativistic electronic structure methods in combination with a relativistic Hamiltonian.

The electron correlation and relativistic effects are not additive [79, 80]. Therefore, an accurate way to treat relativistic effects is to treat electron correlation in combination with a relativistic four or two-component Hamiltonian. Coupled-cluster (CC) methods are widely used for the accurate treatment of dynamical correlation in electronic structure calculations (see [81] or [82] for a review on CC theory). The CC singles doubles, and non-iterative triples (CCSD(T)) method is known to provide reliable results for single-reference molecular systems [83, 84]. Scalar-relativistic (SR) effects can conveniently be included into the non-relativistic CC machinery by replacing the non-relativistic Hamiltonian integrals with relativistic spin-free two or four-component Hamiltonian integrals. Many implementations of SR-CC methods have been developed [85–90] with extensive applications in chemistry [38, 91–95]. Perturbative inclusion of spin-orbit coupling into SR-CC methods has also been developed and applied for accurate calculations for molecules [96, 97]. Implementations of CC methods with non-perturbative treatment of spin-orbit coupling (referred to here as SO-CC methods) have also been developed, including SO-CC methods developed using a four-component DC(B)

Hamiltonian [98–100], quasirelativistic two-component Hamiltonian [101, 102], and effective core potentials (RECP) [103, 104]. These methods have been applied for a variety of atomic and molecular calculations [85, 105–111]. SO-CC methods have larger requirements associated with computational time and storage space than their non-relativistic counterparts. This has limited their applications to relatively small molecules when compared with the non-relativistic and scalar-relativistic CC methods. Two-component methods eliminate the need for small component wavefunction and reduce the number of atomic-orbital (AO) two-electron integrals. This reduces computational time for the HF and integral transformation steps, along with the storage required to store AO integrals. But the requirements of computational time and storage space for CC steps for four-component SO-CC methods are the same as that of two-component SO-CC methods.

The high computational cost in SO-CC methods originates from spin-symmetry breaking. This can be seen from in the MO two-electron integral matrix, with the matrix elements given by

$$\begin{aligned}\langle pq||rs\rangle = &\int d\vec{r}_1 d\vec{r}_2 \chi_p^*(\vec{r}_1)\chi_q^*(\vec{r}_2)\frac{1}{r_{12}}\chi_r(\vec{r}_1)\chi_s(\vec{r}_2) \\ &- \int d\vec{r}_1 d\vec{r}_2 \chi_p^*(\vec{r}_1)\chi_q^*(\vec{r}_2)\frac{1}{r_{12}}\chi_s(\vec{r}_1)\chi_r(\vec{r}_2).\end{aligned} \tag{1.1}$$

Each index $p, q \ldots$ represents a molecular orbital. Each dimension of this tensor doubles because of spin-symmetry breaking with the inclusion of spin-orbit coupling. This leads to a significantly larger MO integral matrix in SO-CC methods compared with their non-relativistic counterparts. For instance, the largest two-electron integral matrix in SO-CC methods, represented by $\langle ab||cd\rangle$ (where $a, b, \ldots$ refer to unoccupied orbitals), requires a storage of about 4 TB for a calculation of about 10-20 atoms compared with 700 GB in its non-relativistic counterpart. SO-CCSD methods are also about 20 times more time-consuming than non-relativistic CC due to loss of spin symmetry. Approaches using scalar-relativistic spin-orbitals in two-component SO-CC methods have been developed [109–114] to reduce computational time and storage space requirements of SO-CC methods. The schemes provide an efficient alternative way to treat spin-orbit coupling in CC methods, but it involves an

approximation related to using orbitals without spin-orbit coupling.

This dissertation develops sparse AO matrix-based algorithms in combination with the X2C Hamiltonian to reduce computational requirements in SO-CC methods and extend the applicability to larger molecules containing heavy elements. The implementation leverages the spin-free nature of AO two-electron integrals and sparsity in AO integral matrix to eliminate the computational bottleneck associated with the storage and evaluation of large MO integral matrices. The AO-based scheme has three main advantages. First, the AO-based scheme avoids the evaluation and storage of large MO integral matrices. Second, it reduces the floating-point operation count of the most time-consuming "ladder term" in SO-CCSD to $\frac{1}{4}$. Third, it allows the use of sparsity in AO integrals to further reduce storage requirements and formal operation count. Note that AO-based algorithms have been extensively exploited in non-relativistic CC methods [115, 116]. The advantages of AO-based algorithms in SO-CC methods are greater than in non-relativistic CC methods, because the AO-based implementation also leads to partial recovery of spin symmetry in SO-CC methods. It is also noteworthy that this development is compatible with all two-component Hamiltonians[74, 117] as well as RECP [118] SO-CC methods. The developments in this dissertation are implemented on the quantum chemistry program package CFOUR [3]. Chapter 2 discusses the theoretical background and chapter 3 discusses the motivation for this work as well as a summary of the present development. Chapter 4 details the development of the sparse AO matrix-based implementation of SO-CC methods to avoid the calculation and storage of four-particle $\langle ab||cd\rangle$ type integrals. Chapter 5 discusses the development of sparse AO matrix-based implementation of the SO-CCSD(T) method to avoid the computation and storage of three-particle $\langle ab||ci\rangle$ type integrals.

The second part of the dissertation, presented in chapter 6, deals with the development of an automatic expression generation library and its application to derive working equations for unitary coupled-cluster (UCC) singles and doubles based third-order polarization propagator theory (UCC3). Implementation of quantum chemistry methods generally involves three steps.

The first step is to derive working equations of the method based on Wick's theorem, either using diagrammatic techniques (see [81]) or string-based manipulation of second quantized operators [119]. The second step deals with combining and reordering the expressions in a form that minimizes the computational cost. The third step deals with implementing the derived working equations. These steps are time-consuming and prone to human error. The automation of these steps thus is highly desirable. Significant effort has been devoted to developing programs to automatically derive and implement working equations in quantum chemistry packages [53, 55, 56, 120–132]. We present a Python-based, open-source and adaptive library (named AutoGen) to carry out the automatic derivation of working equations for single-reference electronic structure methods, e.g. for unitary coupled-cluster (UCC)-based methods. Derivation of working equations for UCC-based methods is more tedious than CC-based methods. Unlike CC methods, a unitary parameterization of wavefunction involves de-excitation operators together with excitation operators in the cluster operator. These de-excitation operators can contract with both the excitation operators and the Hamiltonian. This leads to two main complications. First, the expansion of UCC similarity transformed Hamiltonian becomes non-terminating. Note that the Baker–Campbell–Hausdorff (BCH) expansion of similarity transformed Hamiltonian in CC theory naturally truncates at the fourth order of the commutator. Second, the de-excitation operators increase the complexity of the derivation by increasing the number of intermediates and final unique terms that need to be derived at each order of commutator. Chapter 6 discusses the details of the automatic expression generation library and its application to derive working expressions for the UCC3 method.

Chapter 2

Theoretical background

2.1 Treatment of relativistic effects

2.1.1 Relativistic Hamiltonian

Quantum mechanical equations may be obtained from non-relativistic equations by the use of correspondence principle, where we replace momentum ($\vec{p}$) and energy (E) by quantum operators

$$\vec{p} \rightarrow -i\vec{\nabla}, \qquad E \rightarrow i\frac{\partial}{\partial t}. \tag{2.1}$$

Here we are demonstrating this process for a free particle, where the non-relativistic classical energy of a free particle is given by

$$E = T = \frac{p^2}{2m}, \tag{2.2}$$

which has the kinetic energy (T) of the particle. Note that atomic units are used throughout this chapter, with speed of light c and mass m written explicitly for clarity. Starting with the classical energy equation and using correspondence principle, we derive the time-dependent Schrödinger equation of free particle, given by

$$i\frac{\partial}{\partial t}\Psi(\vec{r},t) = -\frac{1}{2m}\vec{\nabla}^2\Psi(\vec{r},t), \tag{2.3}$$

where $\Psi(\vec{r},t)$ is the time-dependent wavefunction of the particle.

Through a similar process, a free-particle equation of a relativistic quantum particle can

be derived starting with the Einstein's dispersion relation

$$E^2 = \vec{p}^2c^2 + m^2c^4. \tag{2.4}$$

Here, relativistic total energy is written as a sum of relativistic kinetic energy and rest mass energy of a particle with rest mass m. Using replacements from equation 2.1, we get

$$\left(\frac{1}{c}\frac{\partial}{\partial t}\right)^2 \Psi(\vec{r},t) = (\vec{\nabla}^2 - m^2c^2)\Psi(\vec{r},t) \tag{2.5}$$

which can be transformed to write

$$(\vec{\Box} + m^2c^2)\Psi(\vec{r},t) = 0 \tag{2.6}$$

where $\vec{\Box} = (\frac{1}{c^2}\frac{\partial}{\partial t^2} - \vec{\nabla}^2)$. This equation is known as the Klein-Gordon equation. Unlike the Schrödinger equation, this equation is quadratic in time derivative, and as a consequence, the probability density is not positive definite [25].

2.1.1.1 Dirac equation

Dirac was the first to write time-derivative in relativistic free-particle Hamiltonian in a linear form [24] as

$$i\frac{\partial}{\partial t}\Psi(\vec{r},t) = H\Psi(\vec{r},t) = [c\vec{\alpha}\cdot\vec{p} + \beta mc^2]\Psi(\vec{r},t) \tag{2.7}$$

The unknown variables α and β were determined to be such that the energy from the above equation corresponds to the Einstein's dispersion relation, for which the following conditions have to be met:

$$\alpha_i\alpha_j + \alpha_j\alpha_i = \delta_{ij}, \qquad \beta^2 = 1, \qquad \alpha_i\beta = -\beta\alpha_i. \tag{2.8}$$

The conditions in Eq. 2.8 can only be satisfied when α and β are matrices with minimum dimension of 4×4 [133]. The canonical choice of these matrices made by Dirac are

$$\alpha_i = \begin{pmatrix} 0 & \sigma_i \\ \sigma_i & 0 \end{pmatrix}, \qquad \beta = \begin{pmatrix} I_{2\times2} & 0 \\ 0 & -I_{2\times2} \end{pmatrix}, \tag{2.9}$$

where σ_is are conventional Pauli matrices given by

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.10}$$

The equation derived above is referred to as Dirac equation. Substituting wavefunction $\Psi(\vec{r},t) = \Psi(\vec{r})e^{-iEt\hbar}$ and dropping the notation for r dependence, Dirac equation can conveniently be written as

$$\begin{pmatrix} mc^2 & c\vec{\sigma}\cdot\vec{p} \\ c\vec{\sigma}\cdot\vec{p} & -mc^2 \end{pmatrix}\Psi = E\Psi \tag{2.11}$$

or,

$$\left(c\vec{\alpha}\cdot\vec{p} + \beta mc^2\right)\Psi = E\Psi \tag{2.12}$$

Notice that due to the presence of 4×4 matrices, eigenfunctions of Dirac equation have four components. The eigenfunction Ψ is divided into large and small components as

$$\Psi = \begin{pmatrix} \Psi_L \\ \Psi_S \end{pmatrix} \tag{2.13}$$

where

$$\Psi_L = \begin{pmatrix} \Psi_L^\alpha \\ \Psi_L^\beta \end{pmatrix} \qquad \Psi_S = \begin{pmatrix} \Psi_S^\alpha \\ \Psi_S^\beta \end{pmatrix} \tag{2.14}$$

Ψ_L and Ψ_S are referred to as large and small components of the total wavefunction. The large and small component are so named because large component is one to two orders of magnitude larger than small component wavefunction of light elements in the non-relativistic limit (The situation reverses for negative energy solutions) [24]. It should be noted that Dirac equation explicitly include α and β spin of an electron.

The energy eigenvalues of the Dirac equation are

$$E_+ = +\sqrt{p^2c^2 + m^2c^4} \qquad E_- = -\sqrt{p^2c^2 + m^2c^4}, \tag{2.15}$$

corresponding to eigenfunctions Ψ_+ and Ψ_- , respectively. These eigenstates are referred to as positive and negative energy states based on the positive and negative nature of eigenvalues E_+ and E_-, respectively. In the non-relativistic regime, a free particle is expected to have a positive energy. The presence of negative energy states is somewhat counter-intuitive because a particle in a positive energy state could lose energy and spontaneously fall into the negative energy state. Dirac postulated that negative energy states are fully occupied. A vacuum in this interpretation is the state in which all the negative-energy states are filled and all positive-energy states are vacant [24].

2.1.1.2 One and two-electron Hamiltonian

The electrostatic potential due to nucleus is introduced in the Dirac equation to form the one-electron part of the many-body Hamiltonian. This is carried out with the substitution $E = E + \phi_{\text{NUC}}$ where ϕ_{NUC} is the electrostatic potential of the nucleus. Fundamental charge e on an electron is taken as 1 using atomic units. The Born-Oppenheimer approximation is used in electronic structure calculations, which states that the wavefunction of nucleus and electrons can be treated separately in most molecular calculations [134]. This leads to the Dirac equation with nuclear repulsion interaction, given by

$$h_D\Psi = \begin{pmatrix} V & c\vec{\sigma}\cdot\vec{p} \\ c\vec{\sigma}\cdot\vec{p} & V - 2mc^2 \end{pmatrix}\Psi = E\Psi, \tag{2.16}$$

where the energy E is scaled to the non-relativistic energy and $\hat{V} = -e\phi_{NUC}$. In the case of a many-electron system, h_D serves as the one-electron part of the relativistic Hamiltonian.

The two-electron part of the relativistic Hamiltonian is given by [74]

$$H_{12} = \frac{1}{r_{12}} + (h_{Breit})_{12}, \tag{2.17}$$

where the Breit term is defined as

$$(h_{Breit})_{12} = -\frac{\vec{\alpha}_1\cdot\vec{\alpha}_2}{r_{12}} - \frac{(\vec{\alpha}_1\cdot\vec{\nabla}_1)(\vec{\alpha}_2\cdot\vec{\nabla}_2)r_{12}}{2}. \tag{2.18}$$

Notice that the first term in Eq. 2.17 is the Coulomb interaction that describes the charge-charge interaction of the electrons and also arises in the non-relativistic molecular Hamiltonian. The Breit term can be written as a sum of Gaunt term (1st part in Eq. 2.18) and a gauge-dependent term (2nd part in Eq. 2.18) [74]. Coulomb term and Gaunt term give rise to spin-same-orbit (SSO) and spin-other-orbit (SOO) interactions, respectively. The most common choice of the two-electron part of the Hamiltonian is to include Coulomb interaction in the molecular Hamiltonian. This molecular Hamiltonian is referred to as Dirac-Coulomb (DC) Hamiltonian.

2.1.1.3 No-pair Hamiltonian

For the treatment of electron correlation, DC Hamiltonian is projected on only the positive energy states, which gives the "no-pair" DC Hamiltonian [25]. "No-pair" DC Hamiltonian can be written in occupation number representation as

$$\hat{H}_{DC}^{\text{no-pair}} = \sum_{pq}(h_D)_{pq}a_p^\dagger a_q + \frac{1}{4}\sum_{pqrs}(g)_{pq,rs}a_p^\dagger a_q^\dagger a_s a_r, \tag{2.19}$$

where second quantized operators $a_p^\dagger(a_p)$ correspond to creation (annhilation) of an electron in orbital p. $(h_D)_{pq}$ and $(g)_{pq,rs}$ are the one and two-electron integral matrix elements in the relativistic case, while the indices p, $q\ldots$ run only over positive energy states. We can compare this with the non-relativistic Hamiltonian in occupation number representation given by

$$\hat{H} = \sum_{pq}(h)_{pq}a_p^\dagger a_q + \frac{1}{4}\sum_{pqrs}(g)_{pq,rs}a_p^\dagger a_q^\dagger a_s a_r. \tag{2.20}$$

Here $(h)_{pq}$ and $(g)_{pqrs}$ are the one and two-electron integral matrix elements in non-relativistic case and the indices p, $q\ldots$ run over all non-relativistic spin-orbitals. Notice that the only difference in the two lies in the values of one and two-electron integrals. Thus, "no-pair" Hamiltonian takes the same form as the non-relativistic Hamiltonian in occupation number representation. Therefore, quantum chemical machinery can be used for both non-relativistic and relativistic Hamiltonians. Note that the one and two-electron matrix elements in $\hat{H}_{DC}^{\text{no-pair}}$ contain contributions from both large and small components of the wavefunction.

2.1.2 Modified Dirac equation

The block form of the Dirac equation [89] is given as

$$\begin{pmatrix} V & c(\vec{\sigma}\cdot\vec{p}) \\ c(\vec{\sigma}\cdot\vec{p}) & V-2mc^2 \end{pmatrix}\begin{pmatrix} \Psi_L \\ \Psi_S \end{pmatrix} = E\begin{pmatrix} \Psi_L \\ \Psi_S \end{pmatrix}. \tag{2.21}$$

Writing this block equation in the form of two coupled equations in the large and small components, we get

$$V\Psi_L + c(\vec{\sigma}\cdot\vec{p})\Psi_S = E\Psi_L, \tag{2.22}$$

$$c(\vec{\sigma}\cdot\vec{p})\Psi_L + (V-2mc^2)\Psi_S = E\Psi_S. \tag{2.23}$$

The Eq. 2.23 gives the exact coupling operator $\hat{R}$ as

$$\Psi_S = \hat{R}\Psi_L, \qquad \hat{R}(E) = (2mc^2 - V + E)^{-1}c(\vec{\sigma}\cdot\vec{p}), \tag{2.24}$$

which in non-relativistic limit gives

$$\Psi_S = \frac{1}{2mc}(\vec{\sigma}\cdot\vec{p})\Psi_L. \tag{2.25}$$

Implementation of Eq. 2.25 at the basis set level in Dirac equation (Eq. 2.21) is referred to as Kinetic balance condition [135]. This kinetic balance condition can equivalently be implemented by transforming Dirac equation (Eq. 2.21) using the equation

$$\begin{pmatrix} \Psi_L \\ \Psi_S \end{pmatrix} = \begin{pmatrix} I_2 & 0 \\ 0 & \frac{1}{2mc}(\vec{\sigma}\cdot\vec{p}) \end{pmatrix} \begin{pmatrix} \Psi_L \\ \phi_L \end{pmatrix}, \tag{2.26}$$

where ϕ_L is the pseudo-large component [74]. This substitution along with restructuring leads us to the modified Dirac equation, which is given by

$$\begin{pmatrix} V & T \\ T & \frac{W_0}{4m^2c^2} - T \end{pmatrix} \begin{pmatrix} \Psi_L \\ \phi_L \end{pmatrix} = \begin{pmatrix} I_2 & 0 \\ 0 & \frac{T}{2mc^2} \end{pmatrix} \begin{pmatrix} \Psi_L \\ \phi_L \end{pmatrix} E, \tag{2.27}$$

where operators T and W_0 are given by

$$T = \frac{\vec{p}^2}{2m}, \qquad W_0 = \frac{1}{4m^2c^2}(\vec{\sigma}\cdot\vec{p})V(\vec{\sigma}\cdot\vec{p}). \tag{2.28}$$

The operator W_0 can further be divided into a part which is completely spin-free and another which is spin-dependent. This is carried out by using the Dirac identity $(\vec{\sigma}.\vec{A})(\vec{\sigma}.\vec{B}) = \vec{A}.\vec{B} + i\sigma.(\vec{A}\times\vec{B})$ to derive the relation

$$W_0 = \frac{1}{4m^2c^2}(\vec{\sigma}\cdot\vec{p})V(\vec{\sigma}\cdot\vec{p}) \tag{2.29}$$

$$= \frac{1}{4m^2c^2}\vec{p}V\cdot\vec{p} + \frac{1}{4m^2c^2}i\vec{\sigma}\cdot[(\vec{p}V)\times\vec{p}], \tag{2.30}$$

$$= W_0^{SF} + W_0^{SO}. \tag{2.31}$$

This separation of modified Dirac equation into spin-free (W_0^{SF}) and spin-dependent (W_0^{SO}) parts was developed by Dyall in 1994 [89]. Notice that this separation is exact and does not

involve any approximation. The modified Dirac equation can now be written in the form of spin-free and spin-dependent parts as

$$\left[\begin{pmatrix} V & T \\ T & \frac{\vec{p}V\cdot\vec{p}}{4m^2c^2} - T \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \frac{i\vec{\sigma}\cdot[(\vec{p}V)\times\vec{p}]}{4m^2c^2} \end{pmatrix}\right]\begin{pmatrix} \Psi_L \\ \phi_L \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & \frac{T}{2mc^2} \end{pmatrix}\begin{pmatrix} \Psi_L \\ \phi_L \end{pmatrix} E. \tag{2.32}$$

Since our objective is the application of relativistic Hamiltonian on molecular systems, it is convenient to work with a Hamiltonian which reproduces only the positive energy spectrum of the parent four-component Hamiltonian. Several attempts have been made to eliminate the negative energy eigenstates and form the two-component "electrons-only" Hamiltonian [65–78]. The exact two-component (X2C) method has been established as the most promising two-component method [67–69, 74]. In the next subsection, we will discuss the normalized elimination of small component in the X2C scheme.

2.1.3 Exact two-component method

The exact two-component (X2C) method is based on the normalized elimination of small component in the matrix representation of the Dirac equation. The formulation of X2C was proposed by Dyall in 1997 [67] which was revisited and further developed by Kutzelnigg and Liu in 2005 [68]. A correction was made to the renormalization factor by Liu and Peng in 2009 [69]. A production level implementation of the X2C Hamiltonian in the program package Dirac was carried out by Iliaš and Saue [75] in 2007.

The matrix representation of modified Dirac equation (Eq. 2.32) is given by

$$\begin{pmatrix} D^{LL} & D^{LS} \\ D^{SL} & D^{SS} \end{pmatrix}\begin{pmatrix} C^L \\ C^S \end{pmatrix} = \begin{pmatrix} S & 0 \\ 0 & S^{SS} \end{pmatrix}\begin{pmatrix} C^L \\ C^S \end{pmatrix} E, \tag{2.33}$$

where D^{XX} are the matrix elements of the modified Dirac equation in matrix form, S is the overlap matrix of large component and $S^{SS} = \frac{T}{2mc^2}$. C_L and C_S are the basis set expansion coefficients for large and small component wavefuctions, respectively for positive energy states. We define the X matrix as

$$C^S = XC^L. \tag{2.34}$$

In the X2C method, the X matrix is obtained by solving the parent four-component one-electron equation in the matrix form. Using the above relation and applying the matrix $X^\dagger$ on the second equation, we can write the two equations from the Dirac matrix equation as

$$D^{LL}C^L + D^{LS}XC^L = SC^LE, \tag{2.35}$$

$$X^\dagger D^{SL}C^L + X^\dagger D^{SS}XC^L = X^\dagger S^{SS}XC^LE. \tag{2.36}$$

These two equations can be added to give

$$(D^{LL} + D^{LS}X + X^\dagger D^{SL} + X^\dagger D^{SS}X)C^L = (S + X^\dagger S^{SS}X)C^LE, \tag{2.37}$$

which can be written in the form

$$L_+C^L = \tilde{S}C^LE, \tag{2.38}$$

where

$$L_+ = D^{LL} + D^{LS}X + X^\dagger D^{SL} + X^\dagger D^{SS}X, \tag{2.39}$$

$$\tilde{S} = S + X^\dagger S^{SS}X. \tag{2.40}$$

The above equations represent an "electron-only" two component Dirac equation, but it is not ideal in the current form. The "electrons-only" part of the block-diagonalized Dirac equation is expected to be of the form

$$h_+^{FW}C_+^{2c} = SC_+^{2c}E, \tag{2.41}$$

which has S as the metric, instead of $\tilde{S}$ as in Eq. 2.38. The X2C equations are obtained by writing Eq. 2.38 in the form of Eq. 2.41 using a renormalization matrix R as [136]

$$h_+^{FW} = R^\dagger L_+ R, \tag{2.42}$$

$$C^L = RC_+^{2c}. \tag{2.43}$$

The correct form of the matrix R has been derived [69] to be

$$R = (S^{-1}\tilde{S})^{-\frac{1}{2}}. \tag{2.44}$$

X2C theory provides a more convenient form of relativistic many-electron Hamiltonian for molecular calculations. With the derivation of a relativistic two-component Hamiltonian, the remaining task is the development of electronic structure methods using X2C Hamiltonain for accurate treatment of electron correlation. The next subsection discusses electronic structure methods to be used in combination with a relativistic Hamiltonian.

2.2 Treatment of electron correlation

2.2.1 Hartree-Fock theory

Hartree-Fock (HF) wave-function is the simplest antisymmetric wave-function to describe the ground state of a n-electron system [137]. It is a single anti-symmetrized product (single determinant) of occupied orbitals, which can be written for a two-electron system as

$$\Psi_{\mathrm{HF}} = |ij\rangle = \frac{1}{\sqrt{2}} \begin{vmatrix} \phi_i(\vec{r}_1) & \phi_i(\vec{r}_2) \\ \phi_j(\vec{r}_1) & \phi_j(\vec{r}_2) \end{vmatrix} \tag{2.45}$$

$$\equiv \frac{1}{\sqrt{2}}(\phi_i(\vec{r}_1)\phi_j(\vec{r}_2) - \phi_i(\vec{r}_2)\phi_j(\vec{r}_1)). \tag{2.46}$$

Here ϕ_i and ϕ_j are the one-electron wave-functions while $\vec{r}_1$ and $\vec{r}_2$ are the position vectors of the two electrons. Note that we use indices i, j $\ldots$ (a, $b\ldots$) for occupied (unoccupied) molecular orbitals. With the Hamiltonian $\hat{H}$, the HF energy takes the form

$$E_{\mathrm{HF}} = \langle\Psi|\hat{H}|\Psi\rangle = \sum_i \langle i|h|i\rangle + \frac{1}{2}\sum_{ij}(\langle ij|ij\rangle - \langle ij|ji\rangle), \tag{2.47}$$

where the Coulomb and exchange interaction are given by

$$\langle ij|ij\rangle = \int \phi_i^*(\vec{r}_1)\phi_j^*(\vec{r}_2)\frac{1}{|\vec{r}_1 - \vec{r}_2|}\phi_i(\vec{r}_1)\phi_j(\vec{r}_2)d\vec{r}_1 d\vec{r}_2, \tag{2.48}$$

$$\langle ij|ji\rangle = \int \phi_i^*(\vec{r}_1)\phi_j^*(\vec{r}_2)\frac{1}{|\vec{r}_1 - \vec{r}_2|}\phi_j(\vec{r}_1)\phi_i(\vec{r}_2)d\vec{r}_1 d\vec{r}_2, \tag{2.49}$$

respectively. The molecular orbitals (MOs) are taken as a linear combination of atomic orbital basis functions. Self-consistent-field equations are solved to minimise the HF energy. In the X2C-HF method, the MOs (spinors in this case) are written as linear combination of

α and β basis functions as

$$\phi_i = \sum_{\mu} C_{\mu i}^{\alpha} \begin{pmatrix} \chi_\mu \\ 0 \end{pmatrix} + \sum_{\mu} C_{\mu i}^{\beta} \begin{pmatrix} 0 \\ \chi_\mu \end{pmatrix}, \tag{2.50}$$

where $\{\chi_\mu\}$ are scalar basis functions and $\{\begin{pmatrix} \chi_\mu \\ 0 \end{pmatrix}\}$ (and $\{\begin{pmatrix} 0 \\ \chi_\mu \end{pmatrix}\}$) correspond to α (and β) basis functions. The Hartree-Fock method can provide more than 98% of the total energy of a molecule. However, the electron correlation energy, which is defined as the difference between the exact energy and HF energy, is important for calculations of chemical properties. The correlation energy is obtained by including the contributions from excited state determinants into the wavefunction. It is a good time to introduce tools needed in the development of post-HF methods for the computation of correlation energy before going into the discussion of methods for the computation of correlation energy. These tools are the concept of normal ordering of second quantized operators and Wick's theorem.

2.2.2 Basics of electron correlation theory

2.2.2.1 Normal ordering with respect to the physical vacuum

A normal-ordered second quantized operator has all the creation operator to the left of all destruction operators [82]. The ordering simplifies the derivation of matrix elements of operators. This can be understood by looking at the action of creation and annihilation operators on the true vacuum represented by $|0\rangle$. The creation operator acting on vacuum creates a state, as

$$a_q^\dagger a_p^\dagger |0\rangle = |pq\rangle, \tag{2.51}$$

while the action of an annihilation operator on vacuum returns 0 (as there is nothing to annihilate),

$$a_p |0\rangle = 0. \tag{2.52}$$

Using this idea, it can be inferred that all normal ordered string of creation and annihilation operators acting on the vacuum will return 0. For instance,

$$a_p^\dagger a_q^\dagger a_s a_r |0\rangle = 0. \tag{2.53}$$

These second quantized operators have anti-commutation properties, which arise due to the Fermi exclusion principle. These relationships are summarized as

$$[a_p^\dagger, a_q^\dagger]_+ = 0, \qquad [a_p, a_q]_+ = 0, \qquad [a_p^\dagger, a_q]_+ = \delta_{pq}. \tag{2.54}$$

Using these anti-commutation properties, an arbitrary string of second quantized operators can be written as a sum of normal ordered strings and fully "contracted" terms (contractions are defined later). An illustration is presented for an arbitrary operator $\hat{A}$, which is written in second quantized form as

$$\hat{A} = a_p a_q^\dagger a_s a_r^\dagger. \tag{2.55}$$

This operator can be expanded into sum of normal ordered strings as

$$\begin{aligned}
\hat{A} =& \delta_{pq} a_s a_r^\dagger - a_q^\dagger a_p a_s a_r^\dagger \\
=& \delta_{pq}\delta_{rs} - \delta_{pq} a_r^\dagger a_s - \delta_{rs} a_q^\dagger a_p + a_q^\dagger a_p a_r^\dagger a_s \\
=& \delta_{pq}\delta_{rs} - \delta_{pq} a_r^\dagger a_s - \delta_{rs} a_q^\dagger a_p + \delta_{pr} a_q^\dagger a_s - a_q^\dagger a_r^\dagger a_p a_s.
\end{aligned} \tag{2.56}$$

A vacuum expectation value of the operator $\hat{A}$ can be written as

$$\begin{aligned}
\langle 0|\hat{A}|0\rangle =& \langle 0|\delta_{pq}\delta_{rs}|0\rangle - \langle 0|\delta_{pq} a_r^\dagger a_s|0\rangle - \langle 0|\delta_{rs} a_q^\dagger a_p|0\rangle \\
&+ \langle 0|\delta_{pr} a_q^\dagger a_s|0\rangle - \langle 0|a_q^\dagger a_r^\dagger a_p a_s|0\rangle.
\end{aligned} \tag{2.57}$$

All but the first term in the above expression returns 0 as they involve normal ordered string of second quantised operators. As this example illustrates, normal ordering is a useful tool to derive vacuum expectation value of operators in their second quantized form. Next, we will discuss some simple rules to write an arbitrary operator (and product of operators) in second quantized form, into a sum of normal ordered string of operators.

2.2.2.2 Wick's theorem

Anti-commutation relations of second-quantised operators give rise to simple rules to generate normal ordered terms for an arbitraty string of second quantised operators, known as Wick's theorem. The use of Wick's theorem is illustrated using example of the operator $\hat{A}$ discussed in Eq. 2.55.

$$\hat{A} = a_p a_q^\dagger a_s a_r^\dagger, \tag{2.58}$$

$$= \{a_p a_q^\dagger a_s a_r^\dagger\} + \{\overbrace{a_p a_q^\dagger} a_s a_r^\dagger\} + \{\overbrace{a_p a_q^\dagger a_s a_r^\dagger}\} + \{a_p a_q^\dagger \overbrace{a_s a_r^\dagger}\} + \{\overbrace{a_p a_q^\dagger} \overbrace{a_s a_r^\dagger}\}, \tag{2.59}$$

$$= \{a_p a_q^\dagger a_s a_r^\dagger\} + \delta_{pq}\{a_s a_r^\dagger\} + \delta_{pr}\{a_q^\dagger a_s\} + \delta_{rs}\{a_p a_q^\dagger\} + \delta_{pq}\delta_{rs}, \tag{2.60}$$

where contractions, such as $\overbrace{a_p a_q^\dagger}$, are defined by

$$\overbrace{a_p a_q^\dagger} = \delta_{pq}, \qquad \overbrace{a_p^\dagger a_q} = 0, \qquad \overbrace{a_p a_q} = 0, \qquad \overbrace{a_p^\dagger a_q^\dagger} = 0. \tag{2.61}$$

Note that the bracket notation $\{\}$ is used to indicate normal-ordered form of the string written enclosed by it. All pair-wise permutations of contractions are considered in a string of second quantized operators. The number of contractions in each term starts from 0 contractions (giving the normal ordered arrangement of operators) to maximum possible number of contractions.

Wick's theorem can further be generalised to derive normal ordered form of a product of two normal ordered operators. This is illustrated here using an example of a product of two operators in second quantized form, $\hat{A} = \{a_p a_q^\dagger\}$ and $\hat{B} = \{a_s a_r^\dagger\}$ as

$$\hat{A}\hat{B} = \{a_p a_q^\dagger\}\{a_s a_r^\dagger\}, \tag{2.62}$$

$$= \{a_p a_q^\dagger a_s a_r^\dagger\} + \{\overbrace{a_p a_q^\dagger}\{a_s a_r^\dagger\}\} \tag{2.63}$$

$$= \{a_p a_q^\dagger a_s a_r^\dagger\} + \delta_{pr}\{a_q^\dagger a_s\}. \tag{2.64}$$

The idea of all permutations of pairwise contractions remains the same, with the exception that contractions are only permitted between second quantized operators arising from different operators. This generalized form of Wick's theorem is commonly used in the derivation of

matrix elements in electronic structure methods. A problem that arises here is that the number of second-quantized operators involved, if we write the single reference wavefunction using second quantized operators, is large. This is because each occupied orbital is created in true vacuum using a creation operator as

$$|pqrstu\dots\rangle = \dots a_u^\dagger a_t^\dagger a_s^\dagger a_r^\dagger a_q^\dagger a_p^\dagger |0\rangle. \tag{2.65}$$

Even with the help of Wick's theorem, working with such a large string is complicated.

2.2.2.3 Normal ordering with respect to a single slater determinant as redefined vacuum

Simplification is achieved by taking a single determinant as the vacuum instead of working with the true vacuum ($|0\rangle$). Here, all occupied orbitals are represented by indices $i, j, \dots$ and un-occupied orbitals are represented by indices $a, b, \dots$. The new vacuum can be written in terms of true vacuum as

$$\Phi_o = a_i^\dagger a_j^\dagger a_k^\dagger a_l^\dagger \dots |0\rangle, \tag{2.66}$$

where all creation operators are creating occupied orbitals in the true vacuum state. The new contraction definitions with respect to occupied and unoccupied orbitals are

$$\overbrace{a_i^\dagger a_j} = \delta_{ij}, \qquad \overbrace{a_a a_b^\dagger} = \delta_{ab}. \tag{2.67}$$

Other contractions are zero, which are

$$\overbrace{a_i a_j^\dagger} = 0, \qquad \overbrace{a_a^\dagger a_b} = 0. \tag{2.68}$$

The other contractions, that involve mixed occupied and unoccupied orbitals, will be zero as the Kronecker delta function will be zero in that case. All other rules for the Wick's theorem and generalised Wick's theorem remain the same. This is illustrated for a product of two operators as

$$\{a_i^\dagger a_a\}\{a_b^\dagger a_j\} = \{a_i^\dagger a_a a_b^\dagger a_j\} + \{\overbrace{a_i^\dagger a_a}\{a_b^\dagger a_j\}\} + \{a_i^\dagger \overbrace{a_a\}\{a_b^\dagger} a_j\} + \{\overbrace{a_i^\dagger a_a}\{\overbrace{a_b^\dagger a_j}\}\} \tag{2.69}$$

$$= \{a_i^\dagger a_a a_b^\dagger a_j\} + \delta_{ij}\{a_a a_b^\dagger\} + \delta_{ab}\{a_i^\dagger a_j\} + \delta_{ij}\delta_{ab}. \tag{2.70}$$

2.2.3 Coupled-cluster theory

Coupled-cluster (CC) theory uses an exponential parameterization of the wave operator to obtain an approximation to the exact wavefunction [82, 137]. A CC wavefunction is given by

$$|\Psi_{\mathrm{CC}}\rangle = e^{\hat{T}}|\Psi_{\mathrm{HF}}\rangle \tag{2.71}$$

where operator $\hat{T}$ is the cluster operator with weighted excitation operators. The excitation operator $\hat{T}$ can be written as

$$\hat{T} = \hat{T}_1 + \hat{T}_2 + \hat{T}_3 \cdots + \hat{T}_n, \tag{2.72}$$

where each excitation operator creates a series of excited determinants such that electrons from occupied orbitals are excited to un-occupied orbitals. For instance, the action of operator $\hat{T}_2$ on reference wavefunction $|\Psi_{HF}\rangle$, given by

$$\hat{T}_2|\Psi_{HF}\rangle = \frac{1}{4}\sum_{iajb} t_{ij}^{ab} a_a^\dagger a_b^\dagger a_j a_i |\Psi_{HF}\rangle \tag{2.73}$$

creates doubly excited determinants with electrons in occupied orbitals i and j excited to orbitals a and b. t_{ij}^{ab} are the amplitudes associated to the excitations and are determined by the amplitude equations given later (Eq. 2.76).

The action of an exponential cluster operator on the HF wavefunction ensures the important properties of size-extensivity and size-consistency in CC methods at all levels of truncation of the excitation operator [138]. Size-extensivity is the property of a theory that the total energy of a system scales properly with respect to the size of the molecule. Size-consistency is a distinct but related property of a theory that the energy of a super-system, consisting of fragments A and B, is the sum of the fragment energies of A and B in the limit of zero inter-fragment interaction.

The Schrödinger equation can now be written as [82] as

$$\hat{H}e^{\hat{T}}|\Psi_{HF}\rangle = E_{CC}e^{\hat{T}}|\Psi_{HF}\rangle. \tag{2.74}$$

Further, $e^{-\hat{T}}$ can be multiplied on the left along with projection on the reference wavefunction to get the energy equation of CC theory

$$\langle \Psi_{HF} | e^{-\hat{T}} \hat{H} e^{\hat{T}} | \Psi_{\mathrm{HF}} \rangle = E_{CC}. \tag{2.75}$$

The operator $\bar{H} = e^{-\hat{T}} \hat{H} e^{\hat{T}}$ is known as similarity transformed Hamiltonian. Similar to the energy equation, the Eq. 2.74 can be multiplied with the exponential operator $e^{-\hat{T}}$ and projected onto excited determinants to get the amplitude equations

$$\langle \Psi_{\mathrm{I}} | e^{-\hat{T}} \hat{H} e^{\hat{T}} | \Psi_{\mathrm{HF}} \rangle = 0. \tag{2.76}$$

Here, Ψ_{I} are singly, doubly, triply ... excited determinants. The similarity transformed Hamiltonian can be expanded using the Baker–Campbell–Hausdorff (BCH) expansion as

$$e^{-\hat{T}} \hat{H} e^{\hat{T}} = \hat{H} + [\hat{H}, \hat{T}] + \frac{1}{2!}[[\hat{H}, \hat{T}], \hat{T}] + \frac{1}{3!}[[[\hat{H}, \hat{T}], \hat{T}], \hat{T}] + \frac{1}{4!}[[[[\hat{H}, \hat{T}], \hat{T}], \hat{T}], \hat{T}] \tag{2.77}$$

An important property of CC theory is that the BCH expansion of similarity transformed Hamiltonian naturally terminated at the fourth order of commutators. This happens because the operators $\hat{T}s$ are excitation operators and cannot contract among themselves. This leaves the only possible contractions between the $\hat{H}$ and $\hat{T}$ operators. $\hat{H}$ operator can only contract with four operators, which is the number of elementary second quantized operators in the two-electron part of the Hamiltonian (refer to Eq. 2.20). Thus, any fifth-order commutator and beyond will have at least one operator left uncontracted and therefore vanish in the commutator expansion. This natural termination ensures that the CC energy is exact for the order of truncation of excitation operator $\hat{T}$ and no approximation in the BCH expansion is needed. All the commutators of the BCH expansion are computed using Wick's theorem to derive working equations for energy and amplitude equations, which are then re-factorized and implemented in computer programs.

The energy expression for CC theory is given by

$$E_{CC} = E_{HF} + \sum_{ia} f_{ia} t_i^a + \frac{1}{4} \sum_{iajb} \langle ij||ab\rangle t_{ij}^{ab} + \frac{1}{2} \sum_{ijab} \langle ij||ab\rangle t_i^a t_j^b. \tag{2.78}$$

The amplitude equation is dependent on the truncation of excitation operator used. In CCSD method, $\hat{T}$ includes singles and doubles excitations as

$$\hat{T} = \sum_{ia} t_i^a \{a_a^\dagger a_i\} + \frac{1}{4} \sum_{iajb} t_{ij}^{ab} \{a_a^\dagger a_b^\dagger a_j a_i\}. \tag{2.79}$$

The amplitudes equations for CCSD method are too long to write here, and they can be found in the reference [139]. CC approaches, specially CCSD and CCSD augmented with perturbative triple excitations (CCSD(T)), are widely used methods to treat dynamical electron correlation. CCSD(T) method is popularly referred to as the "Gold standard" in computational chemistry.

2.2.4 Equation-of-motion coupled-cluster theory

In equation-of-motion coupled-cluster method (EOM-CC), wavefunction of an excited state (Ψ_{exc}) can be obtained by the action of operator $\hat{R}$ on Ψ_{cc}

$$|\Psi_{exc}\rangle = \hat{R}|\Psi_{CC}\rangle, \tag{2.80}$$

$$= \hat{R}e^{\hat{T}}|\Psi_{HF}\rangle. \tag{2.81}$$

The operator $\hat{R}$ is similar to the configuration interaction [140] method like excitation operator that excites the ground state CC wavefunction to obtain the excited determinants. The operator $\hat{R}$ is given by

$$\hat{R} = \hat{R}_0 + \hat{R}_1 + \hat{R}_2 + \hat{R}_3 + \ldots, \tag{2.82}$$

where $\hat{R}_0$ and $\hat{R}_n$ are defined as

$$\hat{R}_0 = r_0, \tag{2.83}$$

$$\hat{R}_n = \left(\frac{1}{n!}\right)^2 \sum_{ij\ldots} \sum_{ab\ldots} r_{ij\ldots}^{ab\ldots} \{a_a^\dagger a_b^\dagger \ldots a_j a_i\} \tag{2.84}$$

This is an exact parameterization of the excited state wavefunction, given a single reference ground state wavefunction [82]. The equations for determining energy and amplitudes of

EOM-CC methods are obtained by projecting the Schrödinger equation,

$$\hat{H}\hat{R}e^{\hat{T}}|\Psi_{HF}\rangle = E_{exc}\hat{R}e^{\hat{T}}|\Psi_{HF}\rangle, \tag{2.85}$$

onto the ground state, singly excited and doubly excited determinant as

$$\langle\Psi_{HF}|\bar{H}\hat{R}|\Psi_{HF}\rangle = E_{exc}r_o, \tag{2.86}$$

$$\langle\Psi_i^a|\bar{H}\hat{R}|\Psi_{HF}\rangle = E_{exc}\langle\Psi_i^a|\hat{R}|\Psi_{HF}\rangle, \tag{2.87}$$

$$\langle\Psi_{ij}^{ab}|\bar{H}\hat{R}|\Psi_{HF}\rangle = E_{exc}\langle\Psi_{ij}^{ab}|\hat{R}|\Psi_{HF}\rangle. \tag{2.88}$$

Chapter 3

Motivation for efficient implementation of relativistic coupled-cluster methods

3.1 Motivation: the bottleneck of traditional implementation

Computations using coupled-cluster (CC) methods involve a series of tensor contractions to solve amplitude and energy equations. The most time-consuming among the tensor contractions in CC singles and doubles (CCSD) method is the "ladder term" which is given by

$$\frac{1}{2}\sum_{cd}\langle ab||cd\rangle t_{ij}^{cd}. \tag{3.1}$$

This tensor contraction will be taken as an example for the explanation of the atomic orbital (AO)-based scheme in this section. This term involves a contraction between $\langle ab||cd\rangle$-type molecular orbital (MO) integrals and doubles amplitudes t_{ij}^{cd}. Here, a,$b\ldots(i,\ j\ldots)$ refers to unoccupied (occupied) orbitals. These tensors are anti-symmetric with respect to interchange of a with b, c with d, and i with j, for instance, $\langle ab||cd\rangle = -\langle ba||cd\rangle$. $\langle ab||cd\rangle$ is stored for indices $a > b$ and $c > d$ as a matrix of dimension $\frac{N_v(N_v+1)}{2} \times \frac{N_v(N_v+1)}{2}$ and t_{ij}^{cd} is stored for indices $c > d$ and $i > j$ as a matrix of dimension $\frac{N_v(N_v+1)}{2} \times \frac{N_o(N_o+1)}{2}$. Here N_v (N_o) is the number of unoccupied (occupied) orbitals of α spin. This matrix multiplication has a time

Table 3-I. The sizes of large MO integral matrices in non-relativistic, four-component spin-orbit(4c-SO) and two-component spin-orbit(2c-SO) CCSD calculations. This example involves 80 electrons and 1000 virtual spinors (600 basic functions) which refers to a calculation of about 10-20 atoms. $\mu, \nu, \sigma \cdots$ refer to atomic orbital basis functions, $a, b, c \cdots$ refer to unoccupied orbitals while $i, j, k \cdots$ refer to occupied orbitals.

	non-relativistic	4c-SO	2c-SO
$(\mu\nu\|\sigma\lambda)$	120GB	2000GB	120GB
$\langle ab\|\|cd\rangle$	700GB	4000GB	4000GB
$\langle ab\|\|ci\rangle$	144GB	600GB	600GB
$\langle ai\|\|jb\rangle$	75GB	100GB	100GB

complexity of $O(N_v^4 N_o^2)$.

With the inclusion of spin-orbit coupling and resulting loss of spin symmetry, each dimension of $\langle ab||cd\rangle$-type MO integral matrix doubles, i.e, $N_{v,SO}(N_{o,SO}) = 2N_v(N_o)$. Further, the spinors are complex-valued requiring complex matrix multiplication. Complex matrix multiplication is ∼3-4 times more time-consuming than real number matrix multiplication. Overall, the "ladder term" is ∼20 times more time-consuming than in the non-relativistic spin-unrestricted CCSD computation. More importantly, the dimension of $\langle ab||cd\rangle$ matrix increases to $\frac{N_{v,SO}(N_{v,SO}+1)}{2} \times \frac{N_{v,SO}(N_{v,SO}+1)}{2}$. This greatly increases the storage required for $\langle ab||cd\rangle$-type MO integrals in spin-orbit (SO)-CC, for instance, ∼4000 GB storage space is required for an example calculation of about 10-20 atoms (refer to table 3-I) which is much larger than 700 GB that is required in the non-relativistic case. Creation and storage of these large integral matrices becomes a major bottleneck in spin-orbit (SO)-CC calculations.

Inspired by AO-based algorithms in non-relativistic theories [115, 116], we use AO integral matrix to overcome the bottleneck related to handling large MO integral matrices. AOs provide the following advantages over molecular spinors:

1. The size of the AO two-electron integral matrix is smaller than that of the MO two-electron integral matrix in two-component SO-CC case (refer to table 3-I). This is because AOs are pure spin states and AO integrals have an eight-fold permutational

symmetry, i.e, $(\mu\nu|\sigma\lambda) = (\nu\mu|\sigma\lambda) = (\mu\nu|\lambda\sigma) = (\nu\mu|\lambda\sigma) = (\sigma\lambda|\mu\nu) = (\sigma\lambda|\nu\mu) = (\lambda\sigma|\nu\mu) = (\lambda\sigma|\mu\nu)$. As shown in table 3-I, for an example calculation of about 10-20 atoms, AO integrals $(\mu\nu|\sigma\rho)$ require 120 GB storage in two-component SO-CC methods. Even without considering the sparsity in AO integrals, this is much smaller when compared to 4000 GB storage requirement of $\langle ab||cd\rangle$-type MO integrals and 600 GB required by $\langle ab||ci\rangle$-type MO integrals.

2. AO two-electron integrals form a highly sparse matrix [141–143] due to several factors. An important factor contributing to the sparsity is the localized nature of Gaussian functions. In the AO two-electron integral in Mulliken notation

$$(\mu\nu|\sigma\lambda) = \int \chi_\mu^*(\vec{r}_1)\chi_\nu(\vec{r}_1)\frac{1}{|\vec{r}_1 - \vec{r}_2|}\chi_\sigma^*(\vec{r}_2)\chi_\lambda(\vec{r}_2)d\vec{r}_1 d\vec{r}_2, \tag{3.2}$$

the product $\chi_\mu^*(\vec{r}_1)\chi_\nu(\vec{r}_1)$ is a Gaussian function that falls off quickly with the increase of the separation of the centers of χ_μ and χ_ν. Integrals with χ_μ and χ_ν well separated (or χ_σ and χ_λ well separated) are negligible. Other factors also affect the sparsity, including orbital types involved in the integrals (s, p,..), molecular symmetry, etc.

3.2 Summary of the dissertation work on relativistic coupled-cluster methods

In this dissertation, sparse AO matrix-based implementation of exact two-component (X2C) SO-CC methods has been developed. The spin-free nature of AO two-electron integrals and sparsity in the AO integral matrix have been exploited to reduce the storage requirements in SO-CC methods. This implementation extends the applicability of SO-CC methods to treat heavy-element-containing molecules with 10-20 atoms. High-accuracy SO-CC calculations for such large systems is unprecedented. Chapter 4 contains excerpts from our published articles J. Liu, Y. Shen, A. Asthana, and L. Cheng, "Two-component relativistic coupled-cluster methods using mean-field spin-orbit integrals," J. Chem. Phys. **148**, 034106 (2018) and A. Asthana, J. Liu, and L. Cheng, "Exact Two-Component Equation-of-Motion Coupled-Cluster

Singles and Doubles Method Using Atomic Mean-Field Spin-Orbit Integrals," J. Chem. Phys. **150**, 074102 (2019).

The underlying idea of the AO-based algorithms are discussed here using the "ladder term" contraction as an example. The "ladder term" in MO representation is given by $\frac{1}{2}\sum_{cd}\langle ab||cd\rangle t_{ij}^{cd}$. Since spinors are linear combination of AOs, the "ladder term" can be written in terms of AO integrals and partially transformed amplitudes as

$$\frac{1}{2}\sum_{cd}\langle ab||cd\rangle t_{ij}^{cd} = \frac{1}{2}\sum_{\mu\nu\sigma\rho} C_{\mu a}^{*} C_{\nu b}^{*} \langle \mu\nu||\sigma\rho\rangle t_{ij}^{\sigma\rho}, \tag{3.3}$$

where $t_{ij}^{\sigma\rho}$ are partially transformed amplitude term given by

$$t_{ij}^{\sigma\rho} = \sum_{cd} C_{\sigma c} C_{\rho d} t_{ij}^{cd}. \tag{3.4}$$

Detailed expressions, making use of various spin cases of the tensor contractions, have been given in Chapter 2. This transformation converts the matrix multiplication involving $\langle ab||cd\rangle$-type MO integrals into matrix multiplication involving AO two-electron integrals. It should be emphasized that the storage of the AO two-electron integrals, for a calculation of 10-20 atoms, is around 120 GB compared with the size of 4 TB of the corresponding $\langle ab||cd\rangle$-type MO integrals (refer to table 3-I). $\langle ab||cd\rangle$-type MO integral matrix is no longer required to be stored in AO based scheme, which leads to a high reduction in storage requirements in SO-CC methods (refer to table 3-II). In addition to the reduction of storage requirements, the AO-based algorithms also reduce the formal floating-point operation count of the computationally significant "ladder term", in the example in Eq. 3.3, by a factor of four. The number of operation count for "ladder term" in SO-CCSD using MO based algorithm is $\frac{1}{8}N_{v,SO}^{4}N_{o,SO}^{2} \times 4$ (Notice that the factor of 4 has been introduced due to the use complex algebra). The step in AO based algorithm with the highest time complexity is $\sum_{\rho\sigma}\langle \mu\nu||\sigma\rho\rangle t_{ij}^{\sigma\rho}$. The floating point operations of this can be calculated by $\frac{3}{2}N_{ao}^{4}N_{o,SO}^{2}$. Since $N_{v,SO}$ is approximately twice N_{ao}, the AO-based algorithm reduces the floating point operation count by a factor of four [144].

The transformation of the working equations into AO representation has already been established in non-relativistic CC methods [115, 116]. However, the AO-based scheme has

significantly greater benefits in relativistic SO-CC methods, compared with non-relativistic CC methods, due to partial recovery of spin symmetry [144, 145]. Spin symmetry reduces storage requirements of MO integrals as well as computational time for the "ladder term" in non-relativistic CC methods. The computational benefits achieved through the use of spin symmetry are lost with the inclusion of spin-orbit coupling in relativistic CC methods. In an AO-based scheme, the spin-free nature of the AO integrals allows the computation of the "ladder term" in SO-CC to be distributed into various spin cases, partially recovering spin symmetry. Furthermore, the size of MO integrals is much larger in the case of SO-CC methods compared with their non-relativistic counterparts; whereas, the size of the AO integral matrix remains the same in both SO-CC and non-relativistic CC case (refer to table 3-I). Therefore, using AO integrals to avoid large MO integrals is more beneficial in SO-CC methods compared with non-relativistic CC methods. It is also noteworthy that AO-based implementation developed in this dissertation is compatible with all two-component Hamiltonians [74, 117] as well as RECP [118] based SO-CCSD(T) methods.

The focus of the chapter 4 is on the development of sparse AO integral matrix-based algorithms for an efficient implementation of SO-CC methods to leverage sparsity in the AO integral matrix and further reduce storage requirements and formal operation count in SO-CC methods. The SO-CC computations use uncontracted basis sets for an accurate description of the spin-orbit splitting of inner-shell orbitals [146], increasing the number of AO integrals. Further, storing AO integrals has a steep scaling of $O(N_{ao}^4)$, where N_{ao} refers to the number of atomic orbitals. Several schemes have been developed in literature to reduce storage requirements related to two-electron AO integrals, that include density-fitting approaches [147–151], Cholesky decomposition [152–156], direct use of sparsity in tensor contractions involving AO integral matrix [141, 142, 157], etc. An important advantage of using AO-based algorithms is that they allow the use of sparsity in the AO integral matrix to reduce storage requirements and formal operation count. Sparse AO integral matrix-based implementation does not introduce any approximation. As a demonstration, BiH molecule

Table 3-II. The sizes of large MO integral matrices in non-relativistic calculations compared with various implementations of two-component spin-orbit(2c-SO) methods developed in this thesis. This example involves 80 electrons and 1000 virtual spinors (600 basic functions) which refers to a calculation of about 10-20 atoms. $\mu, \nu, \sigma \cdots$ refer to atomic orbital basis functions, $a, b, c \cdots$ refer to virtual orbitals while $i, j, k \cdots$ refer to occupied orbitals.

	non-relativistic	2c-SO	2c-SO $\langle ab\|\|cd\rangle$[a]	2c-SO sparse $\langle ab\|\|cd\rangle$[b]	2c-SO sparse $\langle ab\|\|cd\rangle$ $\langle ab\|\|ci\rangle$[c]
$(\mu\nu\|\sigma\lambda)$	120GB	120GB	120GB	S×120GB[d]	S×120GB[d]
$\langle ab\|\|cd\rangle$	700GB	4000GB	0	0	0
$\langle ab\|\|ci\rangle$	144GB	600GB	600GB	600GB	0
$\langle ai\|\|jb\rangle$	75GB	100GB	100GB	100GB	100GB

[a] 2c-SO with AO based algorithms for terms involving $\langle ab||cd\rangle$-type MO integrals.
[b] 2c-SO with sparse AO matrix based algorithms for terms involving $\langle ab||cd\rangle$-type MO integrals.
[c] 2c-SO with sparse AO matrix based algorithms for terms involving $\langle ab||cd\rangle$ and $\langle ab||ci\rangle$-type MO integrals.
[d] S (sparsity)=fraction of non-zero elements over all elements in AO matrix. Only unique and non-zero values are stored making use of sparsity.

with an un-contracted ANO-RCC basis set only has 6.4% of the unique AO integrals with non-zero values, although molecular symmetry plays an important role in this case. Using sparsity, only 6.4% of the AO integral matrix elements need to be stored, and only 6.4% of the floating-point operations need to be performed. An implementation of SO-CCSD and SO equation-of-motion (EOM)-CCSD methods has been developed that is based on sparse AO integral matrix-based algorithms to avoid $\langle ab||cd\rangle$-type MO integrals (refer to table 3-II for storage requirements for 10-20 atoms). The implementation has been parallelized using the shared memory (OpenMP) based technique. The implementation is developed on the quantum chemistry program CFOUR [3].

In chapter 5, sparse AO matrix-based algorithms for the SO-CCSD with perturbative triples (SO-CCSD(T)) method have been formulated and implemented to avoid the computation of three-virtual index MO integrals ($\langle ab||ci\rangle$). This work is an extension to our work on AO-based algorithms to avoid the creation of $\langle ab||cd\rangle$-type MO integral matrix

for SO-CC computations in chapter 4. Three-virtual index type MO integrals ($\langle ab||ci\rangle$) become the bottleneck in storage requirements when $\langle ab||cd\rangle$-type MO integrals are avoided through AO-based algorithms. For instance, the $\langle ab||ci\rangle$-type MO integral matrix for a SO-CCSD calculation of 1000 virtual orbitals with about 80 electrons correlated, which refers to accurate computation for 10-20 atoms including heavy elements, reaches 600 GB (ref to table 3-I). The overall requirement for such computations surpasses 1 TB of storage space, which is a challenging storage requirement in currently available computational facilities. This implementation reduces the storage requirements of the SO-CCSD(T) method further by half (refer to table 3-II). Overall, this sparse AO matrix-based implementation avoids the creation of $\langle ab||cd\rangle$ and $\langle ab||ci\rangle$-type MO integrals and reduces storage requirements by an order of magnitude compared with the MO-based implementation of SO-CCSD(T). The program has been parallelized using shared-memory intranode (OpenMP) parallelization strategy. The implementation is developed in the quantum chemistry program package CFOUR [3].

Chapter 4

Sparse atomic-orbital (AO) integral matrix-based implementation of spin-orbit coupled-cluster (SO-CC) methods to avoid $\langle ab||cd\rangle$-type molecular orbital (MO) integrals

4.1 Theory

A key advantage of using the AO-based approach in electronic structure methods is that it allows the use of high sparsity in the AO integral matrix to reduce storage requirements and formal floating-point operation count. The AO integrals (in Mulliken notation) are given by

$$(\mu\sigma|\nu\lambda) = \int \chi_\mu^*(\vec{r}_1)\chi_\sigma(\vec{r}_1)\frac{1}{|\vec{r}_1 - \vec{r}_2|}\chi_\nu^*(\vec{r}_2)\chi_\lambda(\vec{r}_2)d\vec{r}_1 d\vec{r}_2, \tag{4.1}$$

where $\vec{r}_1$ and $\vec{r}_2$ represent position of two-electrons, $\frac{1}{|\vec{r}_1-\vec{r}_2|}$ represent relative position of two electrons, and χ_μ represents an atomic orbital wavefunction. Note that anti-symmetrized AO integrals in Dirac notation are

$$\langle\mu\nu||\sigma\lambda\rangle = (\mu\sigma|\nu\lambda) - (\mu\lambda|\nu\sigma). \tag{4.2}$$

The value of integral $(\mu\sigma|\nu\lambda)$ is zero (or close to zero) when orbital pairs given by index (μ,σ) or (ν,λ) have little overlap. This can be due to μ and σ being spatially far apart. The value of the integral can also go to zero with the distance between pairs (μ,σ) and (ν,λ) with $\frac{1}{r_{12}}$.

Several more factors affect integral value and, therefore sparsity, for instance, type of orbitals (s, p, ...), orbital exponent, symmetry in the molecule, cutoff used in evaluating integrals, etc. [143]. A large number of these integrals that have a negligible value make the AO integral matrix a sparse matrix. This sparsity can be used to reduce the storage space required to store the AO integral matrix and formal floating-point operation count of the contractions involving AO integrals. Several AO-based implementations of quantum chemical theories, using sparsity in AO integrals, have been developed in non-rel methods. Janowski et al. [142, 157] in their 2007-08 papers, and Harding et al. in their 2008 paper [141] developed efficient algorithms to use sparsity in AO-based implementation of coupled-cluster methods. Pillio et al. [143] developed a scheme for "ladder term" computation in AO-based algorithms, blocking AO integral matrix into sparse and dense parts for their independent handling.

This chapter discusses an AO-based implementation of SO-CC methods that leverage sparsity in the AO integral matrix to reduce storage requirements and formal operation count in SO-CC methods. The chapter is organised as follows. In section 4.1, we present a formulation for sparse AO integral matrix-based implementation of SO-CC singles and doubles (SO-CCSD) method (4.1.1) and SO equation-of-motion (EOM) CCSD (4.1.2) methods. We analyse the performance of the implementation in section 4.2 and discuss the summary in section 4.3

4.1.1 Working equations for SO-CCSD method

The working equations of the SO-CCSD method take the same form as in the non-relativistic case and are summarized in Appendix I. In this subsection, we will develop a formalism for the implementation of sparse AO integral matrix-based implementation of SO-CCSD methods that avoids the evaluation and storage of four-particle MO integrals. The contraction that involve four-particle MO integrals in SO-CCSD is the "ladder term" given by the tensor contraction $\frac{1}{2}\sum_{cd}\langle ab||cd\rangle t_{ij}^{cd}$. This term is a contraction between four-particle $\langle ab||cd\rangle$-type MO integrals and doubles amplitudes (t_{ij}^{ab}). This term is the most time-consuming term

in SO-CCSD computation. This term scales as $O(N_{v,so}^4 N_{o,so}^2)$ ($N_{v,so}$ and $N_{o,so}$ refers to unoccupied and occupied spinors, respectively) and makes use of the large four-particle MO integral matrix that has a space complexity of $O(N_{v,so}^4)$. We have previously seen in table 3-I that the size of MO integrals become large in SO-CC methods. Storing large matrices on the disk increases disk I/O time drastically, so it is advantageous to store this matrix in memory in quantum chemical packages. An AO-based formulation of SO-CCSD that avoids the evaluation and storage of $\langle ab||cd\rangle$ integrals requires the AO-based formulation of the "ladder term". The "ladder term" can be written in terms of un-transformed AOs as

$$\sum_{cd}\langle ab||cd\rangle t_{ij}^{cd} = \sum_{cd}\sum_{\mu\nu\sigma\lambda} C_{\mu a}^{*} C_{\nu b}^{*} C_{\sigma c} C_{\lambda d} \langle \mu\nu||\sigma\lambda\rangle t_{ij}^{cd}, \tag{4.3}$$

where $\langle \mu\nu||\sigma\lambda\rangle$ are AO two-electron integrals and Cs are MO coefficients. Furthermore, the term can be formulated as

$$\sum_{cd}\langle ab||cd\rangle t_{ij}^{cd} = \sum_{cd}\sum_{\mu\nu} C_{\mu a}^{*} C_{\nu b}^{*} \langle \mu\nu||\sigma\lambda\rangle \sum_{\sigma\lambda} C_{\sigma c} C_{\lambda d} t_{ij}^{cd}, \tag{4.4}$$

$$= \sum_{cd}\sum_{\mu\nu} C_{\mu a}^{*} C_{\nu b}^{*} \langle \mu\nu||\sigma\lambda\rangle t_{ij}^{\sigma\lambda}. \tag{4.5}$$

where $t_{ij}^{\sigma\lambda}$ are partially transformed doubles amplitudes. The evaluation of this term can be done using AO two-electron integrals contracted with partially transformed doubles amplitudes, and further transformation of μ and ν AO indices to MO indices. The term can further be divided into spin-dependent terms as

$$\frac{1}{2}\sum_{\mu\nu\sigma\lambda} C_{\mu a}^{*} C_{\nu b}^{*} \langle \mu\nu||\sigma\lambda\rangle t_{ij}^{\sigma\lambda} = (I^{\alpha\alpha})_{ij,ab} + (I^{\beta\beta})_{ij,ab} + [2(I^{\alpha\beta})_{ij,ab} - 2(I^{\alpha\beta})_{ij,ba}], \tag{4.6}$$

where intermediates (I)s given by

$$(I^{s1,s2})_{ij,ab} = \frac{1}{2}\sum_{\mu\nu} C_{\mu^{s1}a}^{*} C_{\nu^{s2}b}^{*} (\tilde{t}_{\mu\nu,ij}^{s1,s2}), \tag{4.7}$$

$$(\tilde{t}^{s1,s2})_{\mu\nu,ij} = \sum_{\sigma^{s1}\rho^{s2}} \langle \mu^{s1}\nu^{s2}||\sigma^{s1}\lambda^{s2}\rangle t_{ij}^{\sigma^{s1}\lambda^{s2}}; s1, s2 = \alpha, \beta. \tag{4.8}$$

AO integrals are used in the computation of the intermediate $\tilde{t}$.

To take advantage of high sparsity in AO integral matrix, sparse AO integral matrix-based formalism for the above presented AO-based formalism is presented below. The data structure

used for storing sparse AO integral matrix in CFOUR has two variables, one that stores the value of the non-zero unique integral, and the second to store a compact form of its four indices that define its position in the AO integral matrix. This compact representation of matrix elements are then stored in the form of a list. The permutational symmetry in AO integrals plays an important role in storing AO integrals in a compact form, and their use in tensor contractions in AO-based implementations. The permutational symmetry (in Mulliken notation) in AO integrals is given by

$$\begin{aligned}(\mu\sigma|\nu\lambda) &= (\mu\sigma|\lambda\nu),\\ (\mu\sigma|\nu\lambda) &= (\sigma\mu|\nu\lambda),\\ (\mu\sigma|\nu\lambda) &= (\nu\lambda|\mu\sigma).\end{aligned} \tag{4.9}$$

Only one of the eight elements given by these permutational symmetry relationships is stored in the list of unique, non-zero AO integrals.

The implementation to compute "ladder term" using sparse AO integral matrix is based on scalar-vector multiplication, making use of permutational symmetry in AO integrals to generate the intermediate $\tilde{t}$ matrix in Eq. 4.8. Notice here that generation of the $\tilde{t}$ is the step with the highest scaling ($O(N_{v,so}^4 N_{o,so}^2)$) and makes use of the AO integral matrix. To carry out this step using sparsity in AO integrals the list of non-zero and unique AO integrals is iterated over, each element is multiplied with corresponding elements of partially transformed doubles amplitudes based on the permutational symmetry and finally added to corresponding elements of the target $\tilde{t}$ matrix. The construction of the same spin case ($\tilde{t}^{s,s}$) and different spin case ($\tilde{t}^{s1,s2}$) matrices, where $s, s1, s2$ can be α or β and $s1 \neq s2$, is carried out as shown in algorithm 1 and 2, respectively.

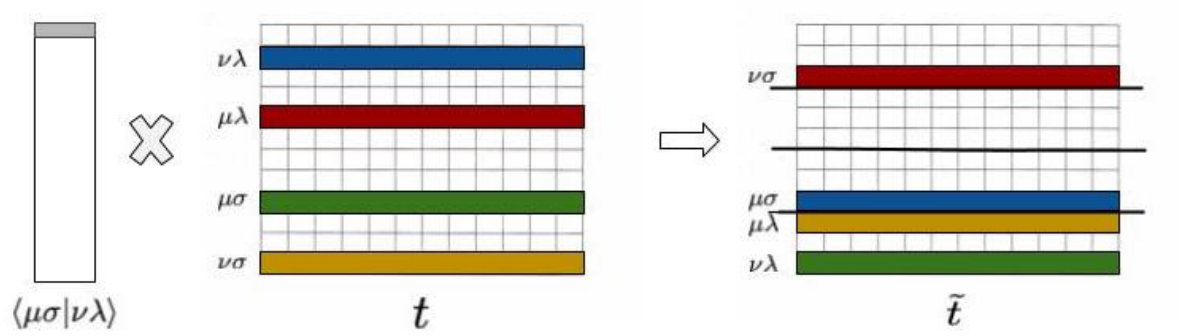

Figure 4-1. Outline of parallelization scheme for sparse AO integral matrix-based implementation of "ladder term". The $\tilde{t}$-matrix divided by extended lines shows the partitioning of this matrix with equal parts of $\tilde{t}$ distributed to different processors. (Represents evaluation of same-spin cases $\tilde{t}^{\alpha\alpha}$ or $\tilde{t}^{\beta\beta}$).

Algorithm 1 Pseudo-code for construction of intermediate ($\tilde{t}^{s,s}$) with $s = \alpha$ or β

A ← List of AO integrals
t ← Matrix of partially transformed doubles amplitudes
procedure MULTIPLY(A,t)
    $\tilde{t}$=0
    for $A_x^{\{\mu\sigma\nu\lambda\}}$ in A **do**
        $\tilde{t}_{(\mu\nu,:)} = \tilde{t}_{(\mu\nu,:)} + A_x^{\{\mu\sigma,\nu\lambda\}} \times t(\lambda\sigma,:)$
        $\tilde{t}_{(\sigma\lambda,:)} = \tilde{t}_{(\sigma\lambda,:)} + A_x^{\{\mu\sigma,\nu\lambda\}} \times t(\mu\nu,:)$
        $\tilde{t}_{(\mu\lambda,:)} = \tilde{t}_{(\mu\lambda,:)} + A_x^{\{\mu\sigma,\nu\lambda\}} \times t(\nu\sigma,:)$
        $\tilde{t}_{(\nu\sigma,:)} = \tilde{t}_{(\nu\sigma,:)} + A_x^{\{\mu\sigma,\nu\lambda\}} \times t(\mu\lambda,:)$

Algorithm 2 Pseudo-code for construction of intermediate ($\tilde{t}^{s1,s2}$) with $s1, s2 = \alpha$ or β and $s1 \neq s2$

A ← List of AO integrals
t ← Matrix of partially transformed doubles amplitudes
procedure MULTIPLY(A,t)
    $\tilde{t}$=0
    for $A_x^{\{\mu\sigma\nu\lambda\}}$ in A **do**
        $\tilde{t}_{(\mu,\nu,:)} = \tilde{t}_{(\mu,\nu,:)} + A_x^{\{\mu\sigma\nu\lambda\}} \times t(\lambda,\sigma,:)$
        $\tilde{t}_{(\sigma,\lambda,:)} = \tilde{t}_{(\sigma,\lambda,:)} + A_x^{\{\mu\sigma\nu\lambda\}} \times t(\mu,\nu,:)$
        $\tilde{t}_{(\mu,\lambda,:)} = \tilde{t}_{(\mu,\lambda,:)} + A_x^{\{\mu\sigma\nu\lambda\}} \times t(\nu,\sigma,:)$
        $\tilde{t}_{(\nu,\sigma,:)} = \tilde{t}_{(\nu,\sigma,:)} + A_x^{\{\mu\sigma\nu\lambda\}} \times t(\mu,\lambda,:)$
        $\tilde{t}_{(\nu,\mu,:)} = \tilde{t}_{(\nu,\mu,:)} + A_x^{\{\mu\sigma\nu\lambda\}} \times t(\sigma,\lambda,:)$
        $\tilde{t}_{(\lambda,\sigma,:)} = \tilde{t}_{(\lambda,\sigma,:)} + A_x^{\{\mu\sigma\nu\lambda\}} \times t(\nu,\mu,:)$
        $\tilde{t}_{(\lambda,\mu,:)} = \tilde{t}_{(\lambda,\mu,:)} + A_x^{\{\mu\sigma\nu\lambda\}} \times t(\sigma,\nu,:)$
        $\tilde{t}_{(\sigma,\nu,:)} = \tilde{t}_{(\sigma,\nu,:)} + A_x^{\{\mu\sigma\nu\lambda\}} \times t(\lambda,\mu,:)$

The parallelization of this implementation is carried out using a shared-memory (OpenMP)-based approach. The target matrix $\tilde{t}$ is partitioned into parts on the row index in the case of $(\tilde{t}^{s1,s2})$ and $(\tilde{t}^{s,s})$ $(s, s1, s2 = \alpha \text{ or } \beta \text{ and } s1 \neq s2)$. The full target matrix $\tilde{t}$ has been shared among processors, each part is then assigned to a specific processor, and the processor iterates over AO integral matrix to compute the assigned part. The memory partitioning is shown in figure 4-1 where batches of the $\tilde{t}$, separated by an extended line in the figure, are assigned to be computed by different processors. It is worthwhile to note that the list of AO integrals was partitioned into batches and processed by different processors in earlier work by Harding et al. [141]. While this strategy leads to easy implementation and efficient division of computational load among processors, this approach is limited by the size of $\tilde{t}$ matrix. In such a strategy, $\tilde{t}$ matrix has to be stored privately in each processor to avoid race condition and added together at the end of computation by the reduction clause in OpenMP. When $\tilde{t}$ matrix becomes large, as in the case of SO-CC methods due to spin-symmetry breaking, storing $\tilde{t}$ matrix in the private memory of each processor can become a limiting factor. For instance, the size of $\tilde{t}$ matrix for an example calculation with 600 basis functions and 70 correlated electrons reaches >1 TB for 48 processors, only to store the $\tilde{t}$ matrix in the private memory of each processor. Partitioning $\tilde{t}$ matrix into parts and assigning parts to different processors eliminates this problem because it is no longer needed to store large $\tilde{t}$ matrix privately in each processor. Therefore, such a scheme is adopted in the present work.

Efficient load-balancing is an important aspect of this implementation. Due to the uneven distribution of sparsity in the AO integral matrix, the computational steps to create different parts of $\tilde{t}$ matrix may be non-uniform. This would translate to a non-uniform distribution of computational load among processors and lead to load imbalance. In this discussion, computational steps that are performed by a processor are referred to as the *load* on that processor. Since the total computational wall time to compute the "ladder term" is governed by the time taken by the slowest processor, delay caused by load-imbalance can have a significant negative impact on the performance of the parallelization. For a good load-balance

in this implementation, a pre-processing step is added that creates the parts of $\tilde{t}$ matrix to be assigned to each processor. In this pre-processing step (referred here as improved load-balancing), the operation count required to calculate each row of $\tilde{t}$ matrix is counted. Then blocks of continuous rows are assigned to continuous processors such that the load on each processor is as close as possible. An outline of the algorithm is given in algorithm 3 for the creation of $(\tilde{t}^{s1,s2})$ with $s1, s2 = \alpha$ or β and discussed here. The first procedure in the algorithm is to count operations required to create each row of $\tilde{t}$ matrix. The operation count to evaluate rows $(\mu\nu)$ have been stored in the list $OperCount$ in the algorithm 3. In the second procedure named $AssignProcessor$, processors have been assigned to rows while continuously counting the total load assigned to each processor. As soon as the load on a processor reaches the $averageload$, the processor index is changed, and no more computations are assigned to that processor. The variable $averageload$ is calculated by dividing the total operation count required for the evaluation of all rows by the number of available processors. The improved load-balancing step is only carried out once before the start of SO-CCSD iterations. It has a scaling of $O(N_{ao}^2)$ which is inexpensive as compared with time-consuming steps in SO-CCSD. The algorithm is expected to distribute the operation count evenly among processors, given the operation count for computing each row of $\tilde{t}$ matrix is small compared with the load on processors.

Algorithm 3 Pseudo-code for improved load-balancing for creation of $\tilde{t}$ matrices in sparse AO integral matrix-based algorithms for the computation of "ladder term"

```
A ← List of AO integrals
MaxProc ← Total number of processors used in the computation
procedure CountOperationsForRows(A)
    for A_x^{{μσ,νλ}} in A do
        case (t̃^{s,s})                                   ▷ same spin case
            OperCount(μν)← +1
            OperCount(σλ)← +1
            OperCount(μλ)← +1
            OperCount(νσ)← +1
        case (t̃^{s1,s2})                                 ▷ s1 ≠ s2, different spin case
            OperCount(μ)← +1
            OperCount(σ)← +1
            OperCount(λ)← +1
            OperCount(ν)← +1
procedure AssignProcessor(A,OperCount,MaxProc)
    AverageLoad ← Total(OperCount)/MaxProc
    CurrProc ← 0
    for z in N_ao/(N_ao*(N_ao+1)/2) do ▷ N_ao and N_ao*(N_ao+1)/2 in cases t̃^{s1,s2} and t̃^{s,s}, respectively
        if Load(CurrProc)≤ AverageLoad then
            Load(CurrProc)← +OperCount(z)
            AssignProc(z) ← CurrProc
        else
            CurrProc← +1
            Load(CurrProc)← +OperCount(z)
            AssignProc(z) ← CurrProc
```

4.1.2 Working equations for SO-EOM-CCSD method

The working equations for SO-EOM-CCSD take the same form as in EOM-CCSD and are summarized in Appendix II. In this subsection, formalism and implementation of sparse AO integral matrix-based algorithms for SO-EOM-CCSD, that avoids the formation of $\langle ab||cd\rangle$-type MO integral matrix and $\bar{H}_{ab,cd}$ matrix, are discussed. The contractions that involve these two matrices are $\frac{1}{2}\sum_{cd}\bar{H}_{ab,cd}r_{ij}^{cd}$ that contributes in the formation of residue vector $\tilde{r}_{ij}^{ab}$ and the term $\sum_f t_i^f \bar{H}_{ab,cd}$ that contributes in the formation of intermediate $\bar{H}_{ab,e,i}$.

First, the AO-based evaluation of the term $\frac{1}{2}\sum_{cd}\bar{H}_{ab,cd}r_{ij}^{cd}$ is developed. The intermediate

$\bar{H}_{ab,cd}$ is given by

$$\bar{H}_{ab,cd} = \langle ab||cd\rangle - P_{ab}\sum_{m} t_m^b \langle am||cd\rangle + \frac{1}{2}\sum_{mn} \tau_{mn}^{ab}\langle mn||cd\rangle, \tag{4.10}$$

where τ is given by

$$\tau_{ij}^{ab} = t_{ij}^{ab} + \frac{1}{2}P_{ij}P_{ab}t_i^a t_j^b. \tag{4.11}$$

In the AO-based implementation of the term $\frac{1}{2}\sum_{cd}\bar{H}_{ab,cd}r_{ij}^{cd}$, $\bar{H}_{ab,cd}$ is not evaluated and stored, and the terms are computed directly as

$$\frac{1}{2}\sum_{cd}\bar{H}_{ab,cd}r_{ij}^{cd} = \frac{1}{2}\sum_{cd}\langle ab||cd\rangle r_{ij}^{cd} - P_{ab}\frac{1}{2}\sum_{mcd} t_m^b\langle am||cd\rangle r_{ij}^{cd} + \frac{1}{4}\sum_{mncd}\tau_{mn}^{ab}\langle mn||cd\rangle r_{ij}^{cd}. \tag{4.12}$$

The first term in equation 4.12 takes the same form as the computation of "ladder term" as discussed in section 4.1. The implementation of this term is carried out using the same sparse AO integral matrix-based formalism that have been developed in equations 4.3 to 4.8 for the "ladder term". The other two terms in equation 4.12 are computed using MO-based approach as

$$-P_{ab}\frac{1}{2}\sum_{mcd} t_m^b\langle am||cd\rangle r_{ij}^{cd} = -P_{ab}\frac{1}{2}\sum_{m}[t_m^b\sum_{cd}(\langle am||cd\rangle r_{ij}^{cd})], \tag{4.13}$$

$$+\frac{1}{4}\sum_{mncd}\tau_{mn}^{ab}\langle mn||cd\rangle r_{ij}^{cd} = +\frac{1}{4}\sum_{mn}[\tau_{mn}^{ab}\sum_{cd}(\langle mn||cd\rangle r_{ij}^{cd})]. \tag{4.14}$$

Notice here that the sequence of brackets define the sequence of computation in this approach. The sequence of operation is chosen to preserve the favourable scaling of $O(N_{v,so}^3 N_{o,so}^3)$ and $O(N_{v,so}^4 N_{o,so}^2)$ for equation 4.13 and 4.14, respectively. The benefit of evaluating the term $\frac{1}{2}\sum_{cd}\bar{H}_{ab,cd}r_{ij}^{cd}$ in this approach is that it avoids the creation and storage of the four-particle index intermediate $\bar{H}_{ab,cd}$.

Second, the AO-based formalism is developed for the term $\sum_f t_i^f \bar{H}_{ab,cd}$ that contributes to the formation of the intermediate $\bar{H}_{ab,e,i}$. As in the previous case we use the expanded form of the intermediate $\bar{H}_{ab,cd}$ in equation 4.10 and write the term $\sum_f t_i^f \bar{H}_{ab,cd}$ as

$$\sum_f t_i^f \bar{H}_{ab,cd} = \sum_f t_i^f\langle ab||cd\rangle - P_{ab}\sum_{fm} t_i^f t_m^b\langle am||cd\rangle + \frac{1}{2}\sum_{fmn} t_i^f\tau_{mn}^{ab}\langle mn||cd\rangle. \tag{4.15}$$

The first term in the equation 4.15 is computed using an sparse AO integral matrix-based formalism as

$$\sum_f t_i^f \langle ab||cd\rangle = \sum_f \sum_{\mu\nu\sigma\lambda} C^*_{\mu a} C^*_{\nu b} C_{\sigma e} C_{\lambda f} \langle \mu\nu||\sigma\lambda\rangle t_i^f, \tag{4.16}$$

$$= \sum_{\mu\nu\sigma} C^*_{\mu a} C^*_{\nu b} C_{\sigma e} \sum_\lambda (\langle \mu\nu||\sigma\lambda\rangle (\sum_f C_{\lambda f} t_i^f)), \tag{4.17}$$

where the sequence of computation has been represented by the sequence of parenthesis in the equation 4.16. This involves creation of partially transformed singles amplitudes, contraction of these partially transformed amplitudes with AO integral matrix, and finally transforming the remaining AO indices to MO indices. The term $\sum_f t_i^f \langle ab||cd\rangle$ can further be written in spin form as

$$\sum_f t_i^f \langle ab||cd\rangle = (\tilde{I'}^{\alpha\alpha})_{ab,e,i} + (\tilde{I'}^{\beta\beta})_{ab,e,i} + [(\tilde{I'}^{\alpha\beta})_{ab,e,i} - (\tilde{I'}^{\alpha\beta})_{ab,e,i}] \tag{4.18}$$

$$+ [(\tilde{I'}^{\beta\alpha})_{ab,e,i} - (\tilde{I'}^{\beta\alpha})_{ab,e,i}]. \tag{4.19}$$

The intermediates $(\tilde{I'})$ are given by

$$(\tilde{I'}^{s1,s2})_{ab,e,i} = \sum_{\mu\nu\sigma} (\tilde{r'}^{s1,s2})_{\mu\nu,\sigma,i} C^*_{\mu^{s1} a} C^*_{\nu^{s2} b} C_{\sigma^{s1} e}; s1, s2 = \alpha \text{ or } \beta, \tag{4.20}$$

where intermediates $(\tilde{r'})$s are given by

$$(\tilde{r'}^{s1,s2})_{\mu\nu,\sigma,i} = \sum_\lambda t_i^{\lambda^{s2}} \langle \mu^{s1}\nu^{s2}||\sigma^{s1}\lambda^{s2}\rangle; s1, s2 = \alpha \text{ or } \beta, \tag{4.21}$$

$$t_i^{\lambda^{s2}} = \sum_a C_{\lambda^s a} t_i^a; s = \alpha \text{ or } \beta. \tag{4.22}$$

A sparse AO integral-based formalism is developed for the computation of the intermediate $(\tilde{r'})$ in the equation 4.21. The creation of the intermediate $(\tilde{r'})$ is computed by the contraction of AO integrals $\langle \mu\nu||\sigma\lambda\rangle$ and the partially transformed amplitudes t_i^λ. As discussed in section 4.1, the value and index of each non-zero AO integral has been stored in the form of a list in CFOUR. Each element of this sparse AO integral list is accessed, multiplied with elements of partially transformed amplitudes t_i^λ, based on permutational symmetry (discribed in Eq. 4.9) and added to the elements of intermediate matrix $(\tilde{r'})$. The pseudo-code for creation

of intermediates ($\tilde{r'}^{s,s}$) and ($\tilde{r'}^{s1,s2}$) ($s, s1, s2 = \alpha$ or β and $s1 \neq s2$) is given in algorithms 4 and 5, respectively. Note that there are a total of 16 contributions based on permuational symmetry in algorithm 5, which are not written in full here for simplicity. They can be easily extrapolated using the given operations. Similar to the "ladder term", parallelization of sparse AO integral matrix-based implemenation of the term $\sum_f t_i^f \langle ab||cd\rangle$ is achieved using a shared-memory (OpenMP)-based approach. The matrix ($\tilde{r'}$) is partitioned on the third index, which creates parts of the matrix ($\tilde{r'}$). The number of parts of ($\tilde{r'}$)x is equal to the number of processors available and each processor is assigned to create a part of the matrix ($\tilde{r'}$). Due to the uneven nature of sparsity in AO integrals, load-balancing is an important aspect of sparse AO integral matrix-based implementation (as discussed above). Improved load-balancing is achieved through a pre-processing step, similar to that in the case of "ladder term", that first counts the operation steps to create each row of ($\tilde{r'}$) matrix, and second it assigns blocks of continuous rows to continuous processors such that the load on each processor is as close to each other as possible. The outline for improved load-balancing for the sparse AO-based implementation of the term $\sum_f t_i^f \langle ab||cd\rangle$ follows the one given for the "ladder term" in algorithm 3.

Algorithm 4 Pseudo-code for construction of intermediate ($\tilde{r'}^{ss}$) with $s = \alpha$ or β

A ← List of AO integrals
t ← Matrix of partially transformed doubles amplitudes
procedure MULTIPLY(A, t)
    $\tilde{t}$=0
    for $A_x^{\{\mu\sigma,\nu\lambda\}}$ in A **do**
        $(\tilde{r'})_{\mu\nu,\sigma,:} = (\tilde{r'})_{\mu\nu,\sigma,:} + A_x^{\{\mu\sigma,\nu\lambda\}} \times t(\lambda,:)$
        $(\tilde{r'})_{\mu\nu,\lambda,:} = (\tilde{r'})_{\mu\nu,\lambda,:} - A_x^{\{\mu\sigma,\nu\lambda\}} \times t(\sigma,:)$
        $(\tilde{r'})_{\sigma\lambda,\mu,:} = (\tilde{r'})_{\sigma\lambda,\mu,:} + A_x^{\{\mu\sigma,\nu\lambda\}} \times t(\nu,:)$
        $(\tilde{r'})_{\sigma\lambda,\nu,:} = (\tilde{r'})_{\sigma\lambda,\nu,:} - A_x^{\{\mu\sigma,\nu\lambda\}} \times t(\mu,:)$
        $(\tilde{r'})_{\sigma\nu,\mu,:} = (\tilde{r'})_{\sigma\nu,\mu,:} + A_x^{\{\mu\sigma,\nu\lambda\}} \times t(\lambda,:)$
        $(\tilde{r'})_{\sigma\nu,\lambda,:} = (\tilde{r'})_{\sigma\nu,\lambda,:} - A_x^{\{\mu\sigma,\nu\lambda\}} \times t(\mu,:)$
        $(\tilde{r'})_{\mu\lambda,\nu,:} = (\tilde{r'})_{\mu\lambda,\nu,:} + A_x^{\{\mu\sigma,\nu\lambda\}} \times t(\sigma,:)$
        $(\tilde{r'})_{\mu\lambda,\sigma,:} = (\tilde{r'})_{\mu\lambda,\sigma,:} - A_x^{\{\mu\sigma,\nu\lambda\}} \times t(\nu,:)$

Algorithm 5 Pseudo-code for construction of intermediate ($\tilde{r'}^{s1s2}$) with $s1, s2 = \alpha$ or β and $s1 \neq s2$

```
A ← List of AO integrals
t ← Matrix of partially transformed doubles amplitudes
procedure MULTIPLY(A, t)
    t̃=0
    for A_x^{μσ,νλ} in A do
        (r̃')_{μ,ν,σ,:} = (r̃')_{μ,ν,σ,:} + A_x^{{μσνλ}} × t(λ,:)
        (r̃')_{μ,ν,λ,:} = (r̃')_{μ,ν,σ,:} − A_x^{{μσνλ}} × t(σ,:)
        ⋮
```

The second and third terms in the equation 4.15 are computed using MO-based formalism as

$$-P_{ab}\sum_{fm} t_i^f t_m^b \langle am||cd\rangle = -P_{ab}\sum_{m} t_m^b \sum_{f}(t_i^f \langle am||cd\rangle), \tag{4.23}$$

$$+\frac{1}{2}\sum_{fmn} t_i^f \tau_{mn}^{ab} \langle mn||cd\rangle = \frac{1}{2}\sum_{mn} \tau_{mn}^{ab} \sum_{f}(t_i^f \langle mn||cd\rangle). \tag{4.24}$$

The computation is carried out as indicated by the sequence of brackets in the equation 4.21 and 4.24.

4.2 Results and discussions

The sparse AO integral matrix-based implementation discussed in this chapter is developed and implemented on a developer version of the CFOUR program package. This implementation will be made available in the later version release of CFOUR program package. The parallelization of the implementation is done using shared memory (OpenMP)-based parallelization. Improved load-balancing algorithms are implemented for more balanced computational load among processors for sparse AO integral matrix-based algorithms. Benchmark studies for this implementation are carried out using large memory nodes on the MARCC supercomputing facility at Baltimore. MARCC facility has 50 large-memory nodes, each with quad Intel "Ivy Bridge" Xeon E7-8857v2 and 1024GB of RAM. An average of 10 wall time values is taken for each wall time measurement point in the graphs. The parallelization tests

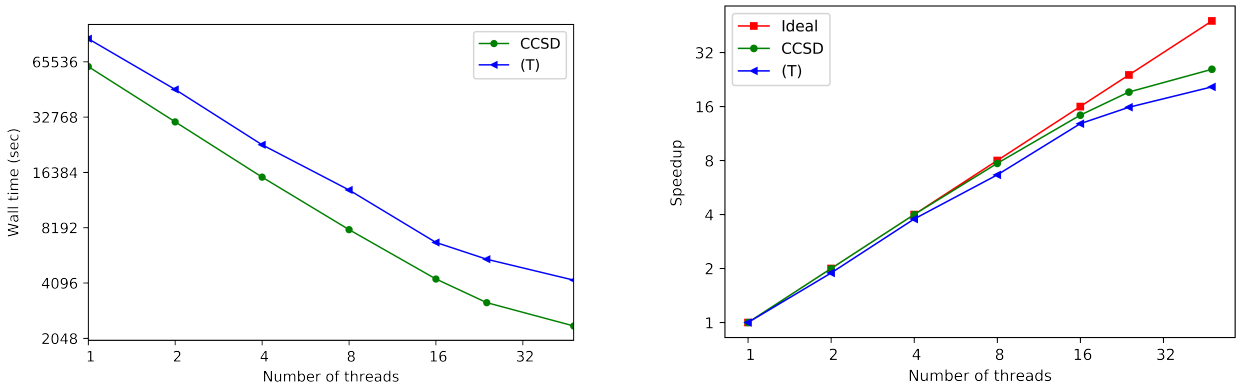

Figure 4-2. Multi-threaded performance of sparse AO integral matrix-based SO-CCSD and (T) methods. The calculation is for BiH molecule using ANO-RCC basis set. The left panel shows elapsed clock time and the right panel shows speedup obtained compared with single thread performance.

have been performed for processors ranging between 1-48 on a single node. The molecule used for benchmarking the implementation is BiH molecule using an uncontracted ANO-RCC basis set. 4f 5d 6s 6p occupied orbitals of Bi atom and 1s in H atom, and virtual orbitals below 1000 Hartree are correlated in the calculations. Overall, there are 432 unoccupied spinors, 30 occupied spinors, and 333 atomic orbitals for this calculation. The sparse AO integral matrix has about 6.4% non-zero unique matrix elements. The calculations presented here have used atomic mean-field (AMF) SO integrals constructed through scalar relativistic HF orbitals that are calculated on spin-free exact-two-component theory in its one-electron variant (SFX2C-1e) [158–160].

Speedup values in the figures 4-2, 4-3 and 4-4 are defined using single core measurement of elapsed time as a reference, where speedup is defined as the ratio of elapsed wall time on multiple cores and that on a single core. The overall implementation of SO-CCSD and (T) calculation scales very well with an increasing number of processors, which can be seen in figure 4-2. The SO-CCSD calculations speed up more than 19 times using 24 processors compared with single-thread performance. The storage requirements are reduced drastically in sparse AO integral matrix-based implementation, first by using AO algorithms to avoid $\langle ab||cd\rangle$-type MO integrals and further by eliminating the need to store full AO integral

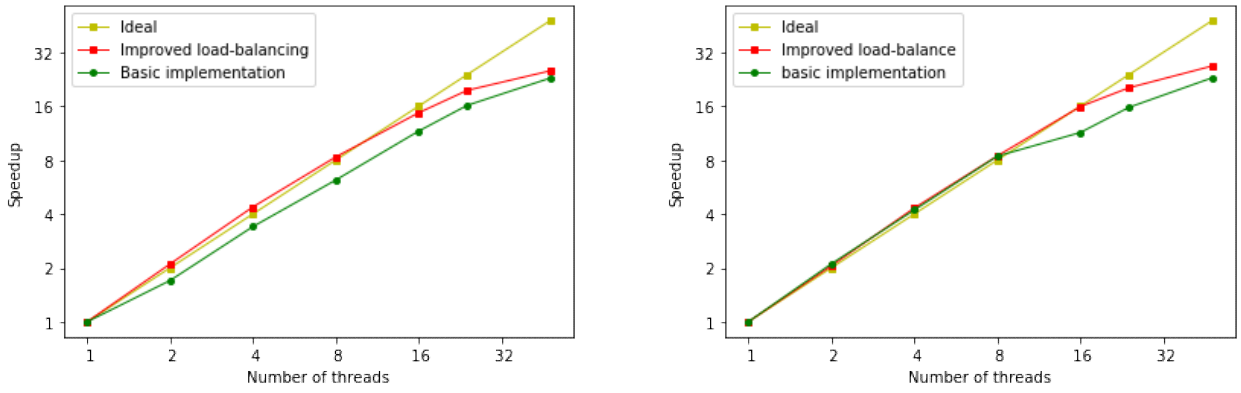

Figure 4-3. Multi-threaded performance for the sparse AO integral matrix-based creation of $\tilde{t}$ matrix, in AO-based algorithm for the "ladder term". The calculation is for BiH molecule using ANO-RCC basis set. The left panel (a) shows shows speedup obtained for the same spin case $\tilde{t}^{s,s}$ and right panel (b) shows speedup obtained for the different spin case $\tilde{t}^{s1,s2}$, where $s, s1, s2 = \alpha$ or β and $s1 \neq s2$. The red curve shows the speedup obtained using improved load-balancing while the green curve shows the speedup obtained using basic implementation.

matrix using sparsity in AO integrals.

The "ladder term" is the most time-consuming term in SO-CCSD, and it is implemented efficiently using sparse AO integral matrix-based algorithms. In our benchmark case, the Ladder term takes about 50% wall time of the CC iterations of SO-CCSD computation of BiH molecule. Therefore, an efficient parallelized implementation of the ladder term is important for efficient SO-CC computations. A parallelized implementation of the ladder term is carried out using the sparse AO integral matrix-based algorithms. The parallel implementation is further enhanced using improved load-balancing algorithms to improve the distribution of computational load among processors on a single node. Improved load-balancing algorithms significantly improve the parallel performance of the "ladder term" as can be seen in figure 4-2. The speedup for the formation of intermediate $\tilde{t}$ increases from about 16 times to more than 19 for the same-spin case $\tilde{t}^{s,s}$ and from about 15 to more than 20 times for the different spin case $\tilde{t}^{s1,s2}$ using 24 processors. The speedup performance reduces slightly while using 48 processors, which may be because of an increase in load-imbalance when using a large number of processors. Perhaps improved algorithms will be needed when a multi-node parallelization

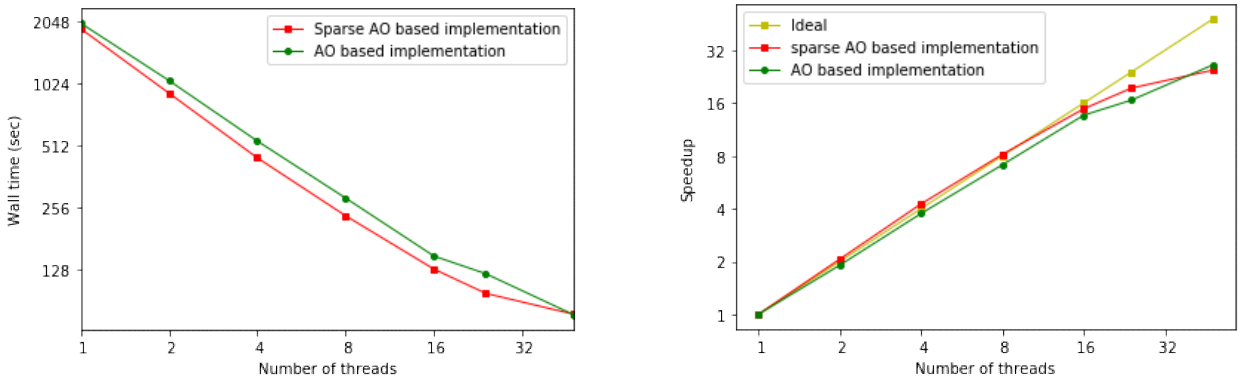

Figure 4-4. Multi-threaded performance of various implementations of the "ladder term". The calculation is for BiH molecule using ANO-RCC basis set. The left panel shows elapsed clock time vs number of threads and the right panel shows speedup obtained compared with single thread performance vs number of threads.

needs to be implemented for the program.

It is worthwhile to note that a super-linear performance has been achieved for 1-8 processors for both the spin cases of the intermediate $\tilde{t}$. This can be seen in figure 4-2 (a) and (b) where the red curve moves above ideal for thread 1-8. For instance, the speedup observed using 8 processors for the same and different spin cases of $\tilde{t}$ intermediate is 8.3 and 8.4 times, respectively. This perceived super-linear speedup might be because of an improved cache optimization when each processor is assigned a part of the matrix $\tilde{t}$ in the parallel algorithm. When the processor handles a smaller part of the matrix, the time spent on fast cache memory re-load due to cache misses reduces. The effective super-linear performance has been observed till about 8 processors. The overall wall time (left panel) and speedup (right panel) of the total "ladder term" is shown in figure 4-3. The red curve shows the performance of sparse AO integral matrix-based implementation using improved load-balancing algorithms, the AO algorithm-based implementation is shown in green. The sparse AO matrix-based implementation performs well in the case of BiH molecule benchmark study. The sparse AO integral matrix-based implementation shows better speedup till 24 threads. It is important to note here that the sparse AO integral matrix-based implementation losses on the highly efficient matrix-matrix multiplication in BLASS subroutines while taking advantage of a

lower formal floating-point operation count for the computation because of sparsity. The performance of sparse AO matrix-based implementation is very similar to the performance of AO-based implementation for BiH molecule example taken. The AO integral matrix has 6.4% non-zero and unique elements that are used in the computation. The performance of sparse AO matrix-based algorithm is likely to improve the performance of AO-based algorithm in molecular calculations where sparsity in the AO integral matrix is larger.

4.3 Summary

A formulation and implementation has been reported for SO-CCSD and SO-EOM-CCSD methods that is based on sparse AO integral matrix-based algorithms to avoid four-particle MO integrals. Sparsity in AO integral matrix has been used to reduce storage required for storing AO integrals and formal floating-point operation count. The implementation has been parallelized using shared-memory parallelization techniques using improved load-balancing algorithms for contractions involving sparse AO integral matrix. The implementation is a part of a developer version of CFOUR and will be available in future release of the code. The implementation extends the applicability of SO-CC methods by reducing storage requirements related to storing a full AO integral matrix using the high sparsity in this matrix. An analysis of the performance of the implementation is reported for 1-48 processors of a single node, and parallel performance of the implementation is studied.

Chapter 5

Sparse atomic-orbital (AO) integral matrix-based implementation of spin-orbit coupled-cluster singles doubles with perturbative triples (SO-CCSD(T)) method to avoid $\langle ab||ci\rangle$-type molecular orbital integrals

5.1 Theory

Strategies to avoid storage of $\langle ab||cd\rangle$ and $\langle ab||ci\rangle$-type MO integrals have been useful in non-relativistic methods [115, 161, 162]. Such a strategy has been particularly important in non-relativistic methods when developing highly-parallel implementation, making use of hundreds of processors, where storage and communication bottlenecks become limiting factors in extending the size of molecular calculations. In such an approach, MO integrals are created when they are required through AO integrals, that themselves are calculated on the fly (integral direct approach) [121, 161–163]. Looking at the increased benefits of AO-based algorithms for the relativistic case due to partial recovery of spin symmetry, we have formulated and implemented sparse AO matrix-based algorithms in this chapter to avoid the evaluation and storage of $\langle ab||ci\rangle$-type MO integrals in addition to $\langle ab||cd\rangle$-type MO integrals. This implementation further extends the applicability of the SO-CCSD(T)

method by reducing the storage requirement of the method. The chapter is divided in the following manner: In section 5.1, we present a theoretical formulation for sparse AO-matrix based algorithms for the SO-CCSD method (5.1.1) and a block-based algorithm to allow the computation of perturbative correction to energy due to triple excitation ((T) computation) in the framework of AO-based implementation (5.1.2). We analyze the performance of the implementation in section 5.2. A summary has been given in section 5.3.

5.1.1 Working equations for SO-CCSD method

SO-CCSD working expressions take the same form as working expressions of spin-dependent CCSD. The SO-CCSD working expressions used in our implementation is presented in detail in Appendix I. The contractions in SO-CCSD that involve $\langle ab||cd\rangle$-type MO integrals are evaluated using AO-based algorithms developed in our earlier work [144] and presented in chapter 4.

In the following, we present a formalism for SO-CCSD(T) method using AO-based algorithms for contractions that involve $\langle ab||ci\rangle$-type MO integral matrix, to avoid the storage of $\langle ab||ci\rangle$-type MO integrals and use of sparsity in AO integrals to avoid storage of full AO integral matrix. The tensor contractions that use the $\langle ab||ci\rangle$-type MO integral matrix arise in the calculation of the intermediates $\tilde{f}_{ae}$, $W_{m,b,e,j}$, and amplitudes r_i^a and r_{ij}^{ab} in SO-CCSD working expressions. These contractions include $\sum_{f,m}\langle am||ef\rangle t_m^f$ contributing to $\tilde{f}_{ae}$, $\sum_f\langle mb||ef\rangle t_j^f$ contributing to $\tilde{W}_{mbej}$, $\sum_{m,f,e}\langle am||ef\rangle t_m^b t_{ij}^{ef}$ and $\sum_c\langle ab||cj\rangle t_i^c$ contributing to r_{ij}^{ab}, and $\sum_{c,b,m}\langle am||bc\rangle t_{mi}^{cb}$ contributing to r_i^a.

First we consider the term $\sum_{f,m}\langle am||ef\rangle t_m^f$ that arise in the formation of the intermediate $\tilde{f}_{ae}$. Intermediate $\tilde{f}_{ae}$ is evaluated by (See Appendix I)

$$\tilde{f}_{ae} = (1-\delta_{ae})f_{ae} - \frac{1}{2}\sum_n f_{me}t_m^a + \sum_{mf} t_m^f\langle ma||fe\rangle - \frac{1}{2}\sum_{mnf}\tilde{\tau}_{mn}^{af}\langle mn||ef\rangle. \tag{5.1}$$

Here $\tilde{\tau}_{ij}^{ab} = t_{ij}^{ab} + \frac{1}{2}(t_i^a t_j^b - t_i^b t_j^a)$ and f is the fock matrix. Evaluation of intermediate $\tilde{f}_{ae}$ requires the computation of the term $\sum_{mf} t_m^f\langle ma||fe\rangle$ [third term in Eq. 5.1], which uses

$\langle ab||ci\rangle$-type MO integral matrix. The underlying idea to develop AO algorithms is that this term can be represented and evaluated in terms of AO two-electron integrals. Spinors being linear combination of AOs, we can write this term as

$$\sum_{f,m}\langle am||ef\rangle t_m^f = \sum_{\mu,\sigma} C^*_{\mu a}C_{\sigma e}(\tilde{r}_{1m})_{\mu,\sigma}, \qquad (\tilde{r}_{1m})_{\mu,\sigma} = (\tilde{r}'_{1m})_{\mu\sigma,m,\lambda}t_m^\lambda - (\tilde{r}'_{1m})_{\mu\lambda,m,\sigma}t_m^\lambda. \tag{5.2}$$

Notice that the intermediates $(\tilde{r}'_{1m})$ has a subscript m that signifies that the intermediate is in Mulliken notation. The intermediate $(\tilde{r}'_{1m})$ and partly transformed amplitude t_m^λ are given by

$$(\tilde{r}'_{1m})_{\mu\sigma,m\lambda} = \sum_\nu (\mu\sigma|\nu\lambda)C^*_{\nu m}, \qquad t_m^\lambda = \sum_f C_{\lambda f}t_m^f. \tag{5.3}$$

The term is evaluated by evaluating the intermediates $(\tilde{r}'_{1m})$ and t_m^λ first, followed by their contraction to produce the intermediate $(\tilde{r}_{1m})$. The intermediate $(\tilde{r}_{1m})$ is finally transformed to MO basis and added to the intermediate $\tilde{f}_{ae}$.

The intermediate $(\tilde{r}'_{1m})$ is evaluated using sparse AO integral matrix. To do this, the list of unique non-zero AO integrals are iterated over element by element, and each element is multiplied with corresponding element of coefficient matrix $C_{\lambda m}$. This intermediate $(\tilde{r}'_{1m})$ is stored for reuse in AO algorithms for the terms $\sum_f\langle mb||ef\rangle t_j^f$, $\sum_{c,b,m}\langle am||bc\rangle t_{mi}^{cb}$ and $\sum_c\langle ab||cj\rangle t_i^c$. The reuse of intermediate $(\tilde{r}'_{1m})$ reduces time-consuming contractions involving sparse AO integral matrix. Storing this intermediate requires a storage space of the order $O(N_v^3N_o)$. For clarity, the algorithm for this sparse-dense type tensor-tensor contraction to create intermediate $(\tilde{r}'_{1m})$ is presented in algorithm 6.

The contributions to the term $\sum_{f,m}\langle am||ef\rangle t_m^f$ can further be divided into contributions by individual spin cases as

$$\sum_{f,m}\langle am||ef\rangle t_m^f = (\tilde{I}_1^\alpha)_{a,e} + (\tilde{I}_1^\beta)_{a,e}. \tag{5.4}$$

$$\tag{5.5}$$

Here intermediate $(\tilde{I}_1)$s are defined as

$$(\tilde{I}_1^s)_{a,e} = \sum_{\mu^s,\sigma^s} C^*_{\mu^s a}C_{\sigma^s e}(\tilde{r}^s_{1m})_{\mu,\sigma}, s = \alpha \text{ or } \beta, \tag{5.6}$$

where the spin dependent intermediates ($\tilde{r}_1$) are defined as

$$\begin{aligned}(\tilde{r}^{s}_{1m})_{\mu,\sigma} = &\sum_{\lambda^s,m}(\tilde{r}'^{s,s}_{1m})_{\mu\sigma,m,\lambda}t^{\lambda^s}_m - \sum_{\lambda^s,m}(\tilde{r}'^{s,s}_{1m})_{\mu\lambda,m,\sigma}t^{\lambda^s}_m \\ &+ \sum_{\lambda^{s2},m}(\tilde{r}'^{s1,s2}_{1m})_{\mu\sigma,m,\lambda}t^{\lambda^{s2}}_m - \sum_{\lambda^{s2},m}(\tilde{r}'^{s2,s1}_{1m})_{\mu\lambda,m,\sigma}t^{\lambda^{s2}}_m,\end{aligned} \tag{5.7}$$

with $s, s1, s2 = \alpha$ or β and $s1 \neq s2$. t^{λ}_m intermediates are defined as

$$t^{\lambda^s}_m = \sum_f C_{\lambda^s f}t^f_m, \tag{5.8}$$

with $s = \alpha$ or β and $(\tilde{r}'^{\alpha,\alpha}_{1m})$, $(\tilde{r}'^{\beta,\alpha}_{1m})$, $(\tilde{r}'^{\alpha,\beta}_{1m})$ and $(\tilde{r}'^{\beta,\beta}_{1m})$ intermediates are defined as

$$(\tilde{r}'^{\beta,\alpha}_{1m})_{\mu\sigma,m,\lambda} = (\tilde{r}'^{\alpha,\alpha}_{1m})_{\mu\sigma,m,\lambda} = \sum_{\lambda^\alpha}(\mu^\alpha\sigma^\alpha|\nu^\alpha\lambda^\alpha)C^*_{\nu^\alpha m}, \tag{5.9}$$

$$(\tilde{r}'^{\alpha,\beta}_{1m})_{\mu\sigma,m,\lambda} = (\tilde{r}'^{\beta,\beta}_{1m})_{\mu\sigma,m,\lambda} = \sum_{\lambda^\beta}(\mu^\beta\sigma^\beta|\nu^\beta\lambda^\beta)C^*_{\nu^\beta m}. \tag{5.10}$$

Algorithm 6 computation of intermediate $(\tilde{r}'_{1m})$

```
(μσ|νλ) ← list of sparse AO integrals, C ← coefficient matrix
procedure (r̃'_1)( (μσ|νλ), C)
    (r̃'_1m) ← 0
    for i=1 to # of elements in (μσ|νλ) do
        (r̃'_1m)_(μσ,m,ν) = Σ_m (μσ|νλ) * C_λ,m
        (r̃'_1m)_(μσ,m,λ) = Σ_m (μσ|νλ) * C_ν,m
        (r̃'_1m)_(σλ,m,σ) = Σ_m (μσ|νλ) * C_μ,m
        (r̃'_1m)_(σλ,m,μ) = Σ_m (μσ|νλ) * C_σ,m
```

Second we consider the term $\sum_f\langle mb||ef\rangle t^f_j$ that contributes to the intermediate $\tilde{W}_{m,b,e,j}$. This contraction can be evaluated using AO-based formalism as

$$\sum_f\langle mb||ef\rangle t^f_j = \sum_{\mu,\nu,\sigma,\lambda}C^*_{\mu m}C^*_{\nu b}C_{\sigma e}C_{\lambda f}\langle\mu\nu||\sigma\lambda\rangle t^f_j, \tag{5.11}$$

$$= \sum_{\mu\sigma}\Big(\sum_\lambda\Big(\sum_\nu(\mu\sigma|\nu\lambda)C^*_{\nu b}\Big)\Big(\sum_f C_{\lambda f}t^f_j\Big)\Big)C^*_{\mu m}C_{\sigma e} \tag{5.12}$$

$$- \sum_{\mu\sigma}\Big(\sum_\lambda\Big((\mu\lambda|\nu\sigma)C^*_{\nu b}\Big)\Big(\sum_f C_{\lambda f}t^f_j\Big)\Big)C^*_{\mu m}C_{\sigma e}. \tag{5.13}$$

As indicated by the sequence of brackets in Eq. 5.13, the contraction is best done by transforming the index f in t^f_j into AO representation and index ν in integral into MO

representation. The two half-transformed intermediate quantities are then contracted together, and finally the resulting intermediate is transformed into MO representation and added to the intermediate $\tilde{W}_{mbej}$. Eq. 5.13 can further be divided into spin cases

$$\sum_f \langle mb||ef\rangle t_j^f = \tilde{I}^{\alpha\alpha}_{m,b,e,j} + \tilde{I}^{\beta\beta}_{m,b,e,j} + \tilde{I}^{\alpha\beta}_{m,b,e,j} + \tilde{I}^{\beta\alpha}_{m,b,e,j}, \tag{5.14}$$

where intermediate quantities represented by $\tilde{I}_{m,b,e,j}$ are

$$\tilde{I}^{s1,s2}_{m,b,e,j} = \sum_{\sigma^{s1},\nu^{s2}} (\tilde{r}^{s1,s2}_{2m})_{\sigma,j,m,\nu} C^*_{\sigma^{s1}b} C_{\nu^{s2}e} \tag{5.15}$$

$$- \sum_{\sigma^{s2},\nu^{s2}} (\tilde{r}'^{s1,s2}_{2m})_{\sigma,\nu,m,j} C^*_{\sigma^{s2}b} C_{\nu^{s2}e}, s1, s2 = \alpha \text{ or } \beta, \tag{5.16}$$

with elements of intermediates $(\tilde{r}'_{2m})^{s1,s2}$ and $(\tilde{r}'_{2m})^{s1,s2}$ given by

$$(\tilde{r}^{s1,s2}_{2m})_{\sigma,j,n,\nu} = \sum_{\lambda^{s1}} (\tilde{r}'^{s1,s2}_{1m})_{\sigma\lambda,m,\nu} t_j^{\lambda^{s1}}, \tag{5.17}$$

$$(\tilde{r}'^{s1,s2}_{2m})_{\sigma,\nu,m,j} = \sum_{\lambda^{s1}} (\tilde{r}'^{s1,s2}_{1m})_{\sigma\nu,m,\lambda} t_j^{\lambda^{s1}}. \tag{5.18}$$

Intermediates $(\tilde{r}'^{s1,s2}_{1m})$ and $t_j^{\lambda^{s1}}$ are defined in Eq. 5.9 and 5.8.

Third we take the term $\sum_{m,f,e} \langle am||ef\rangle t_m^b t_{ij}^{ef}$ that contributes to the t_2 amplitudes. This term can be computed using intermediate $(\tilde{t})_{\mu\nu,ij}$ computed in the "ladder term", that is described in details in our earlier paper [144].

$$(\tilde{t})_{\mu\nu,ij} = \sum_{\sigma,\lambda} \langle \mu\nu||\sigma\lambda\rangle t_{ij}^{\sigma\lambda}, \tag{5.19}$$

where $t_{ij}^{\sigma\lambda} = \sum_{e,f} C_{\sigma e} C_{\lambda f} t_{ij}^{ef}$. The contraction can be written in AO formalism using the intermediate $(\tilde{t})_{\mu\nu,ij}$ as

$$\sum_{m,f,e} \langle am||ef\rangle t_m^b t_{ij}^{ef} = \sum_{\mu,\nu} C^*_{\mu a} C^*_{\nu m} (\tilde{t})_{\mu\nu,ij} t_m^b, \tag{5.20}$$

$$= \sum_{\mu} C^*_{\mu a} \{ \sum_b (\sum_{\nu} C^*_{\nu b} (\tilde{t})_{\mu\nu,ij}) t_m^b \}. \tag{5.21}$$

The contraction can be carried out as represented by the sequence of brackets in Eq. 5.21 by first transforming index ν in $(\tilde{t})_{\mu\nu,ij}$ into MO representation, and contracting the resulting

intermediate with singles amplitudes. Finally, the index μ is transformed into MO representation and added to doubles amplitudes matrix r_{ij}^{ab}. The term can be further represented in spin cases

$$\sum_{m,f,e} \langle am||ef\rangle t_m^b t_{ij}^{ef} = (\tilde{I}_3^{\alpha\alpha})_{a,b,i,j} + (\tilde{I}_3^{\beta\beta})_{a,b,i,j} + 2[(\tilde{I}_3^{\alpha\beta})_{a,b,i,j} - (\tilde{I}_3^{\alpha\beta})_{b,a,i,j}], \tag{5.22}$$

where the intermediates $(\tilde{I}_3^{\alpha\alpha})$, $(\tilde{I}_3^{\beta\beta})$ and $(\tilde{I}_3^{\alpha\beta})$ are

$$(\tilde{I}_3^{s1,s2})_{a,b,i,j} = \sum_{\mu^{s1}} C^*_{\mu^{s2}a} (\tilde{r}_3^{s1,s2})_{\mu,b,ij}, s1, s2 = \alpha \text{ or } \beta, \tag{5.23}$$

with $(\tilde{r}_3^{s1,s2})$ intermediate defined as

$$(\tilde{r}_3^{s1,s2})_{\mu,b,ij} = \sum_m (\tilde{r}_3'^{s1,s2})_{\mu,m,ij} t_m^b, s1, s2 = \alpha \text{ or } \beta, \tag{5.24}$$

and

$$(\tilde{r}_3'^{s1,s2})_{\mu,m,ij} = \sum_{\nu^{s2}} C^*_{\nu^{s2}m} (\tilde{t}^{s1,s2})_{\mu\nu,ij}. \tag{5.25}$$

Fourth we consider the term $\sum_{c,b,m} \langle am||bc\rangle r_{mi}^{cb}$ that contributes to the t_1 amplitudes. The term can be represented in terms of AO integrals as

$$\sum_{c,b,m} \langle am||bc\rangle t_{mi}^{cb} = \sum_{\mu,\nu,\sigma,\lambda} C^*_{\mu a} C^*_{\nu m} C_{\sigma b} C_{\lambda c} \langle \mu\nu||\sigma\lambda\rangle t_{mi}^{cb}, \tag{5.26}$$

$$= \sum_{mu} C^*_{\mu a} \Big(\sum_{\nu} (\mu\sigma|\nu\lambda) C^*_{\nu m} \Big) \Big(\sum_{\sigma,\lambda} C_{\sigma b} C_{\lambda c} t_{mi}^{cb} \Big) \tag{5.27}$$

$$- \sum_{mu} C^*_{\mu a} \Big(\sum_{\nu} (\mu\lambda|\nu\sigma) C^*_{\nu m} \Big) \Big(\sum_{\sigma,\lambda} C_{\sigma b} C_{\lambda c} t_{mi}^{cb} \Big). \tag{5.28}$$

As indicated by the sequence of brackets in Eq. 5.28, the contractions are best done by transforming the index ν in AO integrals into MO representation and indices c and b in t_{mi}^{cb} to AO representation. The two intermediates created are then contracted, and the remaining index is transformed into MO representation and added to singles amplitudes r_i^a. The term can be represented by spin cases as

$$\sum_{c,b,m} \langle am||bc\rangle t_{mi}^{cb} = (\tilde{I}_4^{\alpha})_{a,i} + (\tilde{I}_4^{\beta})_{a,i}, \tag{5.29}$$

where spin intermediates ($\tilde{I}_4$) are given by

$$(\tilde{I}_4^s)_{a,i} = \sum_{\mu^s} C^*_{\mu^s,a}(r_4^s)_{\mu,i}, s = \alpha \text{ or } \beta. \tag{5.30}$$

Intermediates (r_4)s are defined as

$$\begin{aligned}(r_4^{s1})_{\mu,i} = &\sum_{\sigma^{s1},\lambda^{s1},m} (\tilde{r}'^{s1,s1}_{1m})_{\mu\sigma,m,\lambda} t_{mi}^{\lambda^{s1}\sigma^{s1}} - \sum_{\sigma^{s1},\lambda^{s1},m} (\tilde{r}'^{s1,s1}_{1m})_{\mu\lambda,m,\sigma} t_{mi}^{\lambda^{s1}\sigma^{s1}} \\ &+ \sum_{\sigma^{s1},\lambda^{s2},m} (\tilde{r}'^{s1,s2}_{1m})_{\mu\sigma,m,\lambda} t_{mi}^{\lambda^{s2}\sigma^{s1}} - \sum_{\sigma^{s2},\lambda^{s1},m} (\tilde{r}'^{s1,s2}_{1m})_{\mu\sigma,m,\lambda} t_{mi}^{\lambda^{s1}\sigma^{s2}},\end{aligned} \tag{5.31}$$

where intermediates ($\tilde{r}_{1m}$) are defined in Eq. 5.9 and 5.10, and semi-AO transformed amplitudes $t_{mi}^{\lambda\sigma}$ are created by transforming particle indices c and b in doubles amplitude t_{mi}^{cb} into AO indices.

Last we consider the term $\sum_c \langle ab||cj\rangle t_i^c$ that contributes to the doubles amplitudes r_2. This term can be written in AO-based formalism as

$$\begin{aligned}\sum_c \langle ab||cj\rangle t_i^c &= \sum_{\mu,\nu,\sigma,\lambda} C^*_{\mu a} C^*_{\nu b} C_{\sigma c} C_{\lambda j} \langle \mu\nu||\sigma\lambda\rangle t_i^c, \\ &= \sum_{\mu,\nu} C^*_{\mu a} C^*_{\nu b} \Big(\Big(\sum_\lambda (\mu\sigma|\lambda\nu) C_{\lambda j} \Big) \Big(\sum_\sigma C_{\sigma c} t_i^c \Big) \Big) \\ &\quad - \sum_{\mu,\nu} C^*_{\mu a} C^*_{\nu b} \Big(\Big(\sum_\lambda (\nu\sigma|\lambda\mu) C_{\lambda j} \Big) \Big(\sum_\sigma C_{\sigma c} t_i^c \Big) \Big).\end{aligned} \tag{5.32}$$

The contraction is done, as indicated by sequence of brackets in Eq. 5.32, by first transforming index c in t_i^c into AO representation and the index λ in AO integrals into MO representation. The two intermediates formed are contracted together, and finally, all remaining indices are transformed into MO representation and added to doubles amplitude r_{ij}^{ab}. Notice that 8 fold symmetry of AO integrals is used to reuse the same intermediate calculated in Eq. 5.9 and 5.10. The term can be further be divided into spin cases as follows.

$$\sum_c \langle ab||cj\rangle t_i^c = (\tilde{I}_5^{\alpha\alpha})_{ab,ij} + (\tilde{I}_5^{\beta\beta})_{ab,ij} + [(\tilde{I}_5^{\alpha\beta})_{ab,ij} - (\tilde{I}_5^{\alpha\beta})_{ba,ij}] \tag{5.33}$$

$$+ [(\tilde{I}_5^{\beta\alpha})_{ab,ij} - (\tilde{I}_5^{\beta\alpha})_{ba,ij}], \tag{5.34}$$

where the intermediates $(\tilde{I}_5^{\alpha\alpha})_{ab,ij}$,$(\tilde{I}_5^{\beta\beta})_{ab,ij}$,$(\tilde{I}_5^{\alpha\beta})_{ab,ij}$ and $(\tilde{I}_5^{\beta\alpha})_{ab,ij}$ are defined as

$$(\tilde{I}_5^{s,s})_{ab,ij} = \sum_{\mu^s,\nu^s} (\tilde{r}_{5m}^{s,s})_{\mu,i,j,\nu} C^*_{\mu^s a} C^*_{\nu^s b} - \sum_{\mu^s,\nu^s} (\tilde{r}_{5m})^{s,s}_{\nu,i,j,\mu} C^*_{\mu^s a} C_{\nu^s b}, s = \alpha \text{ or } \beta, \tag{5.35}$$

$$(\tilde{I}_5^{s1,s2})_{ab,ij} = \sum_{\mu^{s1},\nu^{s2}} (\tilde{r}_{5m}^{s1,s2})_{\mu,i,j,\nu} C^*_{\mu^{s1} a} C^*_{\nu^{s2} b}, s1, s2 = \alpha \text{ or } \beta, \tag{5.36}$$

Here, the intermediates $(\tilde{r}^{s,s}_{5m})$ and $(\tilde{r}^{s1,s2}_{5m})$ are defined as

$$(\tilde{r}_{5m})^{s1,s2}_{\mu,i,j,\nu} = \sum_{\sigma^{s1}} (\tilde{r}'^{s1,s2}_{1m})_{\mu\sigma,j,\nu} t_i^{\sigma^{s1}}, s1, s2 = \alpha \text{ or } \beta, \tag{5.37}$$

with the intermediate $(\tilde{r}'^{s1,s2}_{1m})$ and t_i^{ν} already defined in Eq. 5.9, 5.10 and 5.8. The pre-calculated intermediate $(\tilde{r}'^{s1,s2}_{1m})$ are reused in this term as well.

5.1.2 Working equations for (T) correction

In this subsection, we present a formulation to compute fourth and leading fifth-order perturbative correction to SO-CCSD energy due to triple excitations $\Big((\mathrm{T})\Big)$ in the AO-based formalism that avoids the creation of MO integrals with more than two particle indices. The equations take the same form as in the spin-dependent CCSD(T) method. (T) correction to energy is given by

$$E_{(T)} = \sum_{i>j>k} \sum_{a>b>c} t(c)^{abc}_{ijk} D^{abc}_{ijk} (t(c)^{abc}_{ijk} + t(d)^{abc}_{ijk}). \tag{5.38}$$

Here $t(c)^{abc}_{ijk}$ is the connected term which arises from fourth order correction to energy from triple excitation while $t(d)^{abc}_{ijk}$ is the disconnected term which arises from the leading fifth order correction to energy from triple excitation. The terms $t(c)^{abc}_{ijk}$ and $t(d)^{abc}_{ijk}$ include all the permutations of indices a, b and c and i,j and k as

$$t(c)^{abc}_{ijk} D^{abc}_{ijk} = (W_c)^{abc}_{ijk} - (W_c)^{acb}_{ijk} + (W_c)^{bca}_{ijk}, \tag{5.39}$$

$$t(d)^{abc}_{ijk} D^{abc}_{ijk} = (W_d)^{abc}_{ijk} - (W_d)^{acb}_{ijk} + (W_d)^{bca}_{ijk}, \tag{5.40}$$

where

$$\begin{aligned} (W_c)^{abc}_{ijk} = &\sum_d \langle ab||di\rangle t^{cd}_{jk} + \sum_l \langle jk||lc\rangle t^{ab}_{li} \\ &- \sum_d \langle ab||dj\rangle t^{cd}_{ik} - \sum_l \langle ik||lc\rangle t^{ab}_{lj} \\ &+ \sum_d \langle ab||dk\rangle t^{cd}_{ji} + \sum_l \langle ji||lc\rangle t^{ab}_{lk}, \end{aligned} \tag{5.41}$$

$$(W_d)^{abc}_{ijk} = -\langle ab||jk\rangle t^c_i + \langle ab||ik\rangle t^c_j - \langle ab||ji\rangle t^c_k, \tag{5.42}$$

$$D^{abc}_{ijk} = f_{ii} + f_{jj} + f_{kk} - f_{aa} - f_{bb} - f_{cc}. \tag{5.43}$$

Computation of (T) correction suitable for OpenMP-based shared-memory parallelization have been carried out using "ijkabc" algorithm [164, 165]. Several implementations have been developed using "ijkabc" algorithm [121, 161] that calculates W and V for all "abc" indices with fixed "ijk" indices through contractions involving doubles amplitudes t_{ij}^{ab}, $\langle ab||ci\rangle$-type and $\langle ij||ka\rangle$-type MO integrals as represented in Eq. 5.41 and 5.42. Using these intermediates, $t(c)_{ijk}^{abc}$ and $t(d)_{ijk}^{abc}$ are computed as in Eq. 5.39 and 5.40 and finally (T) correction contribution is computed for fixed "ijk" values as in Eq. 5.38. This method computes contributions to (T) correction from one set of "ijk" indices at a time and avoids storing full six index large arrays such as $t(c)_{ijk}^{abc}$. An outline of these steps are shown in algorithm 7.

Algorithm 7 MO-based algorithm to compute (T) correction

integrals ← MO integral matrices, f ← fock matrix, t_{ij}^{ab},t_i^a ← singles and doubles amplitudes matrices.
procedure TRIPLES(integrals, f,t_{ij}^{ab},t_i^a)
    (T) =0
    for i=1 to $N_{o,so}$ **do**
        for j=i to $N_{o,so}$ **do**
            for k=j to $N_{o,so}$ **do**
                compute $(W_c)_{ijk}^{abc}$ and $(W_d)_{ijk}^{abc}$ for fixed index i,j and k from Eq. 5.41 and 5.42.
                $D_{ijk}^{abc} = f_{ii} + f_{jj} + f_{kk}$
                for a=1 to $N_{v,so}$ **do**
                    for b=a to $N_{v,so}$ **do**
                        for c=b to $N_{v,so}$ **do**
                            $D_{ijk}^{abc} = D_{ijk}^{abc} - f_{aa} - f_{bb} - f_{cc}$
                            compute $t(c)_{ijk}^{abc}$ and $t(d)_{ijk}^{abc}$
                            $E_{(T)} = E_{(T)} + t(c)_{ijk}^{abc} D_{ijk}^{abc} (t(c)_{ijk}^{abc} + t(d)_{ijk}^{abc})$

In the framework of AO-based algorithms that avoid the creation of $\langle ab||cd\rangle$ and $\langle ab||ci\rangle$-type MO integrals, $\langle ab||ci\rangle$-type MO integrals are not pre-computed and available to evaluate the intermediate $(W_c)_{ijk}^{abc}$ and $(W_d)_{ijk}^{abc}$. The contributions to $(W_c)_{ijk}^{abc}$ from Eq. [5.41] can be

written as

$$(W_c)^{abc}_{ijk} \leftarrow \sum_d \langle ab||di\rangle t^{cd}_{jk} + \sum_l \langle jk||lc\rangle t^{ab}_{li}, \tag{5.44}$$

$$(W_c)^{abc}_{ijk} \leftarrow -\sum_d \langle ab||dj\rangle t^{cd}_{ik} - \sum_l \langle ik||lc\rangle t^{ab}_{lj}, \tag{5.45}$$

$$(W_c)^{abc}_{ijk} \leftarrow \sum_d \langle ab||dk\rangle t^{cd}_{ji} + \sum_l \langle ji||lc\rangle t^{ab}_{lk}. \tag{5.46}$$

Since $(W_c)^{abc}_{ijk}$ is to be constructed for fixed values of indices i, j and k at a time, the first contribution appearing in Eq. [5.41] can be formulated as a matrix multiplication as

$$\sum_d \langle ab||di\rangle t^{cd}_{jk} \equiv A1^i(ab,d)t^{jk}(d,c), \tag{5.47}$$

where $A1^i(ab,d)$ represents $\langle ab||ci\rangle$ MO integral matrix with a fixed value of index i and $t^{jk}(d,c)$ represents t^{dc}_{jk} with fixed index j and k. Since $\langle ab||ci\rangle$-type MO integral matrix is not pre-evaluated and stored in the AO-based formalism, $A1^i(ab,d)$, $A1^j(ab,d)$ and $A1^k(ab,d)$ should be made available at each iteration of the loop in "ijk". These integrals may be evaluated in three main ways. First, the three integrals, $A1^i(ab,d)$, $A1^j(ab,d)$ and $A1^k(ab,d)$, may be evaluated at each iteration of i, j and k. Second, full matrix of $\langle ab||ci\rangle$ may be evaluated before the start of the subroutine. Third, a part of $\langle ab||ci\rangle$ matrix may be evaluated and made it available for the (T) computation at a time. Evaluating $A1^i(ab,d)$, $A1^j(ab,d)$ and $A1^k(ab,d)$ at each iteration of i, j and k or "on the fly" method uses the least amount of storage space (three matrices of size $O(N^3_{v,so})$ where ($N_{v,so}$ refers to number of unoccupied spinors) but has a very high floating point operation count overhead. Construction of full $\langle ab||ci\rangle$-type MO integral matrix before starting the evaluation process for (T) correction which has the least floating point count overhead but has the worst storage complexity which is $O(N^3_{v,so}N_{o,so})$ $N_{v,so}(N_{o,so})$ refers to number of unoccupied(occupied) spinors).

A block-based algorithm has been formulated and implemented that evaluates blocks of $\langle ab||ci\rangle$ at a time and computes contribution to (T) correction that arises from evaluated blocks. The blocks of the matrix for $\langle ab||ci\rangle$-type integrals are defined by dividing the index i in $\langle ab||ci\rangle$ into parts. The number of these blocks depends on the available memory, which

is described later. We may create one, two, or three blocks at a time. The creation of one block refers to the creation of a full $\langle ab||ci\rangle$ matrix before the start of (T) computation. The block-size, l, in this case is $N_{o,so}$ and this full $\langle ab||ci\rangle$ matrix can only be created if enough storage space is available. On the other hand, the creation of three blocks at a time is a flexible way of storing only a part of the $\langle ab||ci\rangle$ matrix at any given time. The block size (l) can be controlled by changing the range of i indices in the block of the matrix $\langle ab||ci\rangle$. Block-size (l) can be assigned such that an optimal strategy is implemented based on the amount of storage space available in the computation.

Algorithm 8 block-based algorithm to compute (T) correction

$B \leftarrow$ a block of $\langle ab||ci\rangle$-type integrals
compute$(B_1, B_2, B_3) \leftarrow$ evaluates contributions to (T) with elements from B_1,B_2 and B_3
procedure BLOCKING(B)
    for x=1 to $\frac{N_{o,SO}}{S}$ **do**
        evaluate block B_x
        compute(B_x, B_x, B_x) ▷ all contributions from B_x
        for y=x to $\frac{N_{o,SO}}{S}$ **do**
            evaluate block B_y
            compute(B_x, B_x, B_y) ▷ contributions from combination of B_x and B_y
            compute(B_x, B_y, B_y)
            for z=y to $\frac{N_{o,SO}}{S}$ **do**
                evaluate block B_z
                compute(B_x, B_y, B_z) ▷ contributions from combination of B_x, B_y and B_z

The block-based algorithm for calculating (T) correction is shown in algorithm 8. The algorithm starts by computing the first block of $\langle ab||ci\rangle$-type MO integral matrix B_x. This block of $\langle ab||ci\rangle$-type MO integral matrix is computed as

$$(B_x)_{ab,c,i'} = \sum_{\mu,\nu,\sigma,\lambda} C^*_{\mu a} C^*_{\nu b} C_{\sigma c} C_{\lambda i'} \langle \mu\nu||\sigma\lambda\rangle, \tag{5.48}$$

$$= \sum_{\mu,\nu,\sigma} C^*_{\mu a} C^*_{\nu b} C_{\sigma c} \Big(\sum_{\lambda} (\mu\sigma|\lambda\nu) C_{\lambda i'} \Big) \tag{5.49}$$

$$- \sum_{\mu,\nu,\sigma} C^*_{\mu a} C^*_{\nu b} C_{\sigma c} \Big(\sum_{\lambda} (\nu\sigma|\lambda\mu) C_{\lambda i'} \Big). \tag{5.50}$$

Here i' is the hole index in the block x. Notice that block-size l is number of i' indices in any

block by definition. The equation can be written in terms of spin cases as

$$B_x(ab,c,i') = (\tilde{I}_6^{\alpha\alpha})_{ab,c,i'} + (\tilde{I}_6^{\beta\beta})_{ab,c,i'} + [(\tilde{I}_6^{\alpha\beta})_{ab,c,i'} - (\tilde{I}_6^{\alpha\beta})_{ba,c,i'}] + [(\tilde{I}_6^{\beta\alpha})_{ab,c,i'} - (\tilde{I}_6^{\beta\alpha})_{ba,c,i'}], \tag{5.51}$$

where the intermediates $(\tilde{I}_6^{\alpha\alpha})_{ab,c,i'}$, $(\tilde{I}_6^{\alpha\beta})_{ab,c,i'}$, $(\tilde{I}_6^{\beta\alpha})_{ab,c,i'}$ and $(\tilde{I}_6^{\beta\beta})_{ab,c,i'}$ are

$$(\tilde{I}_6^{s,s})_{ab,c,i'} = \sum_{\mu^s,\sigma^s} (r_{6m}^{s,s})_{\mu\sigma,b,i'} C^*_{\mu^s,a} C_{\sigma^s,c} - \sum_{\mu^s,\sigma^s} (r_{6m}^{s,s})_{\nu\sigma,a,i'} C^*_{\nu^s,b} C_{\sigma^s,c}, \tag{5.52}$$

$$(\tilde{I}_6^{s1,s2})_{ab,c,i'} = \sum_{\mu^{s1},\sigma^{s1}} (r_{6m}^{s1,s2})_{\mu\sigma,b,i'} C^*_{\mu^{s1},a} C_{\sigma^{s1},c}, \tag{5.53}$$

where $s, s1, s2 = \alpha$ or β and $s1 \neq s2$. Spin intermediates (r_{6m}) are defined as

$$(r_{6m}^{s1,s2})_{\mu\sigma,b,i} = \sum_{\nu^{s2}} (r'^{s1,s2}_{6m})_{\mu\sigma,\nu,i} C^*_{\nu^{s2},b}, s1, s2 = \alpha \text{ or } \beta, \tag{5.54}$$

with

$$(r'^{s1,s2}_{6m})_{\mu\sigma,\nu,i} = \sum_{i'} (\mu\sigma|\nu\lambda) C_{\lambda^{s2},i'}. \tag{5.55}$$

$(W_c)_{ijk}^{abc}$ is then computed using Eq. 5.41-5.42 for all i, j and k in the block B_x. Contributions to (T) originating from this block are then evaluated using the $(W_c)_{ijk}^{abc}$ matrix, similar to the MO-based algorithm (lines 12, 13 and 14 in algorithm 7). Based on the block-size (l), B_y and B_z blocks may be evaluated using Eq. 5.49 through 5.55. Then, contributions arising from all i, j and k indices present in these blocks are evaluated. Notice that three blocks will be created for $l < \frac{N_{o,so}}{2}$, two for $\frac{N_{o,so}}{2} \leq l < N_{o,so}$ and one for (case of full $\langle ab||ci\rangle$ matrix) $l = N_{o,so}$. Finally, all contributions are evaluated that arise when indices i, j and k exist in different blocks.

An important aspect of the block-based algorithm is the choice of the block-size (l). A strategy has been employed for choosing a block-size (l) such that the algorithm to compute (T) correction does not require any more storage space than the SO-CCSD steps. By analysis of the storage spaces needed for the SO-CCSD program in the CFOUR package, it is concluded that the storage space allocated for storing DIIS vectors and temporary memory for matrix manipulation can be suitably used to store the blocks of $\langle ab||ci\rangle$-type MO

integral matrix and intermediate ($\tilde{r}_6$) in Eq. 5.55. DIIS vectors are no longer needed after the SO-CCSD calculation is converged, and temporary memory arrays are also available at the triples calculation for use. Therefore these are ideal storage places that can be used for block-based algorithm. Something to take into account here is that the overhead of evaluation of these blocks does not decrease in the range $\frac{N_{o,so}}{2} > l > N_{o,so}$. Taking into consideration this information, we have employed the strategy to select a suitable block-size (l) shown in algorithm **??**. Essentially, this is a scheme which selects $l = N_{o,so}$ or $l = \frac{N_{o,so}}{2}$ if enough storage space is available to store blocks of $\langle ab||ci\rangle$-type MO integral matrix of this size. If enough storage space is not available, the highest block size possible is selected in the range $0 < l < \frac{N_{o,so}}{2}$. Further, an analysis of scaling of block-based algorithm and estimate of block-size for relevant computational scenarios is carried out in section 5.2.

Algorithm 9 Strategy to select block-size (l)

```
if storage of full ⟨ab||ci⟩-type MO integral matrix is possible then
    l ← N_{o,so}
else
    if l = N_{o,so}/2 is possible then
        l ← N_{o,so}/2
    else
        l ← highest possible value based on available storage space.
```

5.2 Results and discussions

The above-described implementation of SO-CCSD(T) methods has been implemented in a developer version of CFOUR and will be made available in the later versions of the program. Benchmarking calculations and performance analysis are carried out on large-memory nodes of the MARCC computing facility. MARCC facility has 50 large-memory nodes each with quad Intel "Ivy Bridge" Xeon E7-8857v2 and 1024GB of RAM. OpenMP parallelization is used to parallelize the program using efficient load balancing algorithms (discussed in an earlier chapter) for sparse AO matrix-based algorithms. The benchmarking was done with the number of threads ranging from 1 to 48 on a single node. The calculations presented use

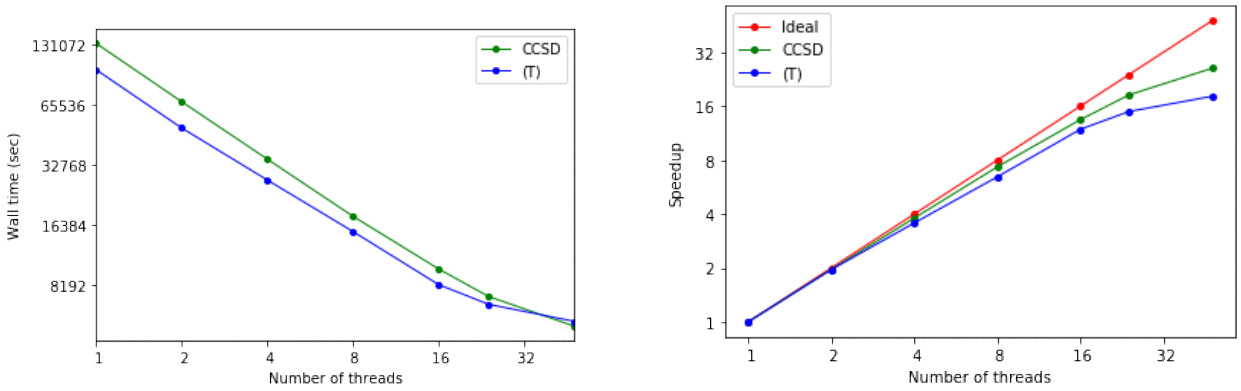

Figure 5-1. Multi-threaded performance for SO-CCSD and (T) computation for BiH molecule using ANO-RCC basis set ($N_{o.so} = 30$, $N_{v,so} = 432$ and $N_{ao} = 333$). The left panel shows wall-clock time and the right panel shows speedup obtained compared with single thread performance.

atomic mean-field (AMF) SO integrals constructed through scalar relativistic HF orbitals that are calculated on spin-free exact-two-component theory in its one-electron variant (SFX2C-1e) [158–160]. The benchmarking calculations for performance analysis of the code are carried out on BiH molecule with 30 correlated electrons. The basis set used in the calculations is ANO-RCC. The ratio of $N_{o,so}/N_{v,so}$ is 0.07. The electrons in the orbitals 4f 5d 6s 6p of Bi atom and 1s in H atom, and virtual orbitals below 1000 Hartree are correlated in the calculations. The sparsity of the AO integral matrix is 6.4% for this calculation. The performance of the multi-threaded implementation of AO-based algorithms driven implementation of SO-CCSD(T) method is analyzed using multiple threads on a single node in figures 5-1 and 5-2. Further, analysis of (T) algorithm is presented in figures 5-3 and 5-4. Speedup values of the right panel of this figure and the figures in this chapter are obtained using the single-core measurement as reference. The speedup is defined as the ratio of elapsed wall time on multiple cores and elapsed wall time on a single core.

The figure 5-1 shows the elapsed wall time (left panel) and speedup (right panel) of SO-CCSD and (T) calculation. The overall speedup of CCSD computation is excellent for CPU-based parallelization on a single node. The SO-CCSD computation speeds up as much as 18.5 times, while the (T) calculation speeds up by 15 times by using 24 cores. However, efficiency seems to somewhat drop when using a higher number of cores for both SO-CCSD

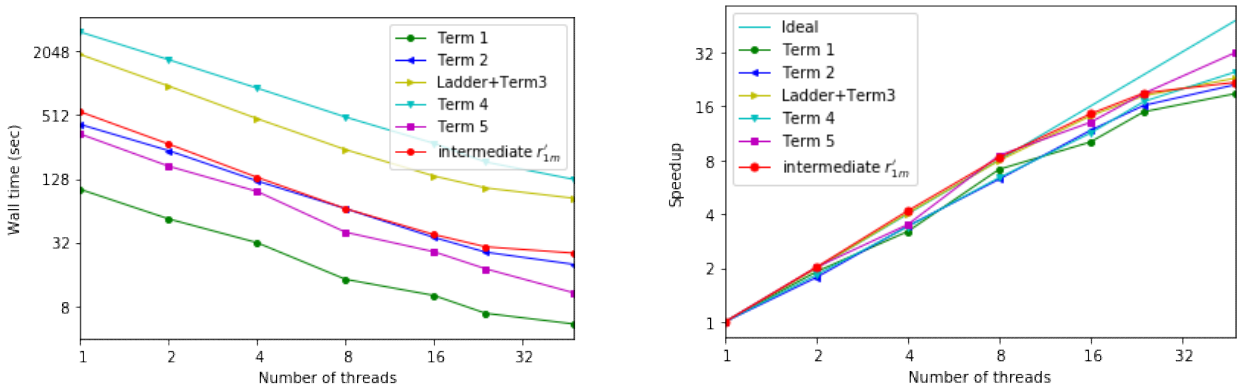

Figure 5-2. Multi-threaded performance for various contractions in SO-CCSD that use AO-based algorithms. The calculation is for BiH molecule using ANO-RCC basis set. The left panel shows elapsed clock time and the right panel shows speedup obtained compared with single thread performance. Here, term 1, term 2, term 3, term4 and term 5 refers to $\sum_{f,m}\langle am||ef\rangle t_m^f$, $\sum_f\langle mb||ef\rangle t_j^f$, $\sum_{m,f,e}\langle am||ef\rangle t_m^b t_{ij}^{ef}$, $\sum_c\langle ab||cj\rangle t_i^c$, and $\sum_{c,b,m}\langle am||bc\rangle t_{mi}^{cb}$, respectively.

and (T) computation. Further, an analysis of multi-threaded performance is presented for parts of SO-CCSD in figure 5-2. This figure shows the elapsed time and speedup for terms that are implemented using AO-based algorithms and formation of intermediate $\tilde{r}'_{1m}$ through sparse AO integral matrix. The term 3, which consists of one of the highest scaling "Ladder term" (scales as$O(N^6)$ where N is the number of occupied or un-occupied orbitals) in SO-CCSD, takes about $\frac{1}{4}^{\text{th}}$ of the elapsed time of SO-CCSD iterations. Further, $\tilde{r}'_{1m}$ intermediate (see Eq. 5.9 and 5.10) formation is a time taking step that scales as $O(N_{sparse}N_{o,so})$, where N_{sparse} is the number of non-zero, unique elements in sparse AO integral matrix. In the worst case scenario, this term scales as $O(N_{ao}^4 N_{o,so})$. This intermediate is formed by contraction between sparse AO integral matrix and coefficient matrix. The algorithms are parallelized by OpenMP and an efficient load-balancing algorithm is implemented to provide a uniform computational load to different threads, as described in earlier chapter (chapter 4). This intermediate is created once for each SO-CCSD iteration and reused in AO-based algorithms for the terms $\sum_{f,m}\langle am||ef\rangle t_m^f$, $\sum_f\langle mb||ef\rangle t_j^f$, $\sum_{c,b,m}\langle am||bc\rangle t_{mi}^{cb}$, and $\sum_c\langle ab||cj\rangle t_i^c$. The elapsed time for this intermediate formation depends on the sparsity of AO integrals, along with the total number of atomic-orbitals (N_{ao}) and the total number of electrons correlated ($N_{o,so}$). The sparsity in AO integral matrix for the BiH molecule used in the computation is 6.4%. The

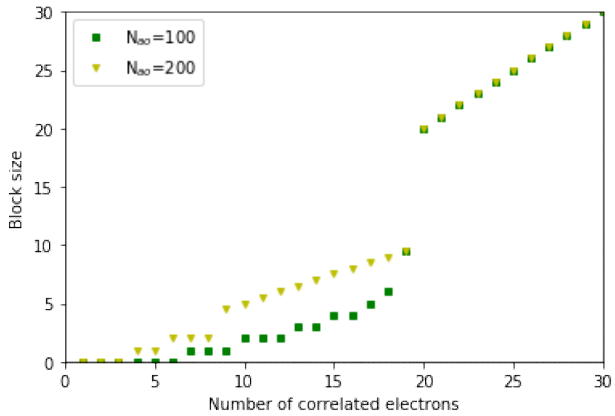

Figure 5-3. Block-size allocated using available storage space vs number of correlated electrons ($N_{o,so}$). The block-size is calculated using algorithm 8. The sum of the total number of unoccupied spinors ($N_{v,so}$) and occupied spinors ($N_{o,so}$) is kept as 200.

elapsed time and speedup for the formation of the intermediate $\tilde{r}'_{1m}$ is given in the red line in figure 5-2. As seen in the figure, a super-linear speedup performance is observed for this intermediate for the first few to several threads. For instance, the speedup for 8 threads is 8.25 times. This super-linear speedup observed may be because of a reduction in cache misses when this task is divided among different processors. The parallelization algorithm for this term assigns the creation of parts of the final intermediate matrix $\tilde{r}'_{1m}$ to different threads. With multiple threads, the size of the target matrix, that a processor needs to access, reduces. More localized memory access reduces cache misses. This effective super-linear speedup observation lasts till 8 threads in our calculation. All terms with AO-based algorithm performs very well considering an OpenMP-based parallelization for a single node.

An analysis of the block-based algorithm for the calculation of (T) correction, described in section 5.1.2, has been presented here. The algorithm can be divided into two parts: the creation of blocks of $\langle ab||ci\rangle$-matrix, and the calculation of (T) correction to energy. The former is carried out in lines 5, 8, and 12 in algorithm 8, while the latter retains the same structure as in MO-based implementation and is carried out in lines 6, 9, 10, 13 in algorithm 8. The creation of blocks of the $\langle ab||ci\rangle$-matrix is an overhead in this algorithm as it is not present in the MO-based implementation given in algorithm 7. This overhead

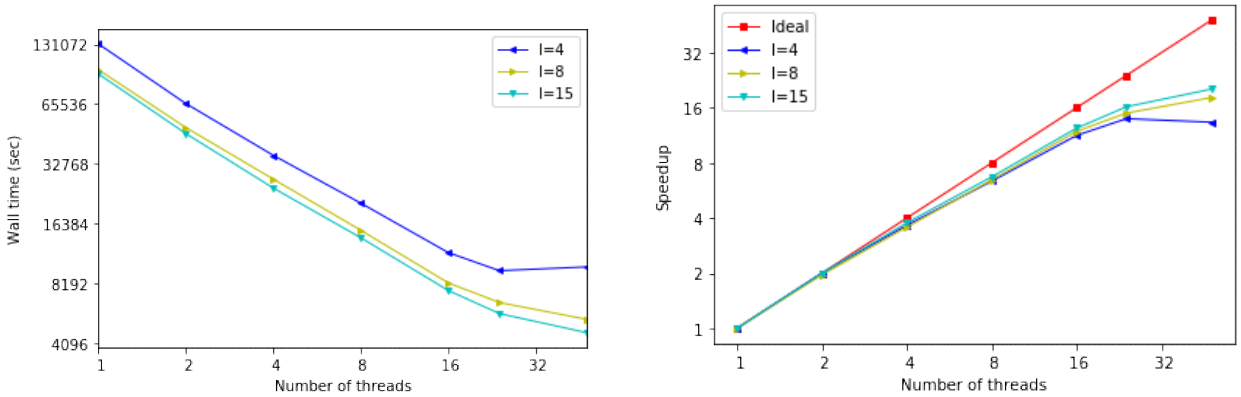

Figure 5-4. Multi-threaded performance of block-based evaluation of (T) correction for BiH molecule using ANO-RCC basis set as a function of block-size (l). The left panel shows wall clock time and the right panel shows speedup obtained as a function of number of threads on a single node.

scales as $\mathrm{O}(\frac{N_v^4 N_o^3}{l^2})$ where l is the block size of $\langle ab||ci\rangle$-matrix where index i is divided into blocks. For comparison, the scaling of the evaluation of the (T) correction to energy scales as $\mathrm{O}(N_v^4 N_o^3)$. It is clear from the scaling that the overhead related to the creation of blocks of the $\langle ab||ci\rangle$-matrix reduces fast with increasing block-size l. A numerical plot to examine automatically allocated block-sizes for different cases that may arise in chemical computations is presented in figure 5-3. The figure plots block-size allocated with number of correlated electrons ($N_{o,so}$) in hypothetical calculations. The plot shows that the allocated block-size reaches $\frac{N_o}{2}$ for ratio $\frac{N_{o,so}}{N_{v,so}} \approx 0.1$ without considering any frozen orbitals. A larger block-size reduces the overhead cost associated with the creation of blocks of the $\langle ab||ci\rangle$-matrix. The block-size increases with increase in ratio of $\frac{N_o}{N_v}$ and as more orbitals are frozen (case of $N_o + N_v < 2N_{ao}$). Note that the total number of correlated electrons are kept fixed to 200 ($N_{o,so} + N_{v,so} = 200$) and the total number of atomic-orbitals are 100 and 200 in green and yellow markers, respectively. This numerical analysis is based on the strategy to select block-size presented in algorithm 9, and the implementation in CFOUR where the storage space dedicated DIIS vectors and temporary matrix operations are used to store the blocks of the $\langle ab||ci\rangle$-matrix. Figure 5-4 shows elapsed time and speedup for (T) computation for different values of block-size l for BiH molecule with ANO-RCC basis set. Generally, a larger

value of block-size l leads to better serial and parallel performance of the (T) algorithm.

5.3 Summary

A formulation and implementation of the SO-CCSD(T) method has been reported that is based on sparse AO-matrix-based algorithms to avoid the creation of four-particle index $\langle ab||cd\rangle$-type and three-particle index $\langle ab||ci\rangle$-type MO integrals. This implementation improves computational efficiency by avoiding the creation and storage of large MO two-electron integrals. Sparsity in AO integrals reduces the storage required for AO integrals and the formal floating-point operation count in time-consuming tensor contractions in SO-CCSD(T) method. The implementation is efficiently parallelized using shared-memory (OpenMP)-based parallelization, where improved load-balancing algorithms are used for contractions involving sparse AO integral matrix. The implementation has been implemented on a developer version of CFOUR program package and will be made available in later release. A computational analysis of the implementation is reported.

Chapter 6

Automatic expression generation for unitary coupled-cluster (UCC)-based methods

6.1 Introduction

The complexity of several classes of modern quantum chemistry methods has reached a stage where manual derivation of working equations and implementation of these equations on scientific softwares have become a limiting factor in testing many new computational methods. As a result, there has been a significant effort to automate the derivation of expressions and code development for quantum chemistry methods by exploiting the repetitive nature of the work involved in this process. A pioneer effort to do automatic derivation of expressions for electronic structure methods was carried out by Paldus and Wong in the 1970s. They derived nBody MBPT theory expressions using Feynman diagrams of the Hugenholtz type [120] automatically [122, 166]. An early fully functional automatic code generator to derive expressions, refactor and write code for electronic structure methods was developed by Janssen and Schaefer which was named SQSYM (Second Quantization SYmbol Manipulator) program [123]. The program derived expressions and reorganized them in a manner that reused intermediates, which were directly interpreted and executed using the program CORR. This code was used in the implementation of a spin-adapted high-spin open-shell CCSD method.

Automatic expression and code generator programs have proved very useful in modern electronic structure methods, a review can be found by Hirata [57]. Kállay and co-workers developed a diagram-based automatic generator program [55] which was used in the development of coupled-cluster methods with excitation operators truncated at an arbitrary order [167]. Hirata and co-workers developed the powerful automatic program generator Tensor Contraction Engine (TCE) in 2004 [124, 125] which has extensively been used in the development of configuration-interaction, many-body perturbation theory, and coupled-cluster methods for ground and excited states [57]. General implementations of configuration-interaction and active space coupled-cluster methods have been carried out by Olsen through automatic equation and code generation [168]. General implementations of relativistic methods have also been developed using automatic generators by Hirata and coworkers in 2007 [169], and Nataraj, Kllay, and Visscher in 2010 [170]. More recent developments in this area include a generally applicable automatic expression and code development software applicable to fermionic and bosonic contractions by Zhao and Scuseria [127, 171–173] and an expression and code generator tool for the implementation of electronic structure methods on graphics processing units (GPUs) and Tensor Hyper Contraction (THC) based electronic structure methods by Song and co-workers [56, 174, 175]. There are several more notable developments in automatic expression and code generation [58, 126, 176–179].

Automatic generators have been extensively helpful in the development of multi-reference methods. Multi-reference problem is especially difficult to deal with using hand-derived diagrams and hand-written software due to both, the number of terms involved in the methods and the complexity of normal ordering for active spaces (such as the Kutzelnigg and Mukherjee [180] type normal ordering). Automatic generator tools have proved essential for the development of internally contracted (ic) multi-reference (MR) methods [53, 130, 181]. Gecco program was developed and used by Hanauer and co-workers [53] for the development of ic-MRCC methods. Nooijen and co-workers have developed the code Automatic Program Generator (APE) which is used in the development of similarity-transformed equation-of-

motion (STEOM-)CC and state selective (SS) MR method developments [182, 183]. The newer developments of this program make use of Kutzelnigg and Mukherjee type normal ordering and has been used in the development of SS-EOM-CC [184] and MR-EOM-CC methods [129, 185, 186]. Canonical transformation (CT) theory developed by Neuscamman and co-workers has made extensive use of automatic derivation of expressions [54, 128]. Due to the unitary parameterization of the wavefunction, the number of unique terms involved in CT methods is exceptionally large. For instance, the number of unique terms in commutator $[[\hat{H}, \hat{A}], \hat{A}]_{1,2}$, where $\hat{A} = \hat{T} + \hat{T^{\dagger}}$ is 16,935 [54, 128]. Note that the natural truncation of similarity transformed Hamiltonian using a Baker–Campbell–Hausdorff (BCH) expansion that occurs at the fourth-order of commutator in single-reference coupled-cluster methods, does not happen in CT and ic-MRCC theory-based methods.

In this chapter, we present automatic expression generation library (named AutoGen) [187] that is especially well suited for the development of unitary coupled-cluster (UCC) based methods; along with the application of Autogen to derive working equations for UCC singles and doubles based third-order polarisation propagator theory (UCC3) [188]. The equations of UCC3 presented in this chapter are from our publication J. Liu, A. Asthana, L. Cheng and D. Mukherjee, "Unitary coupled-cluster based self-consistent polarization propagator theory: A third-order formulation and pilot applications," J Chem. Phys. **148**, 244110 (2018). The AutoGen library is an open-source, Python-based library that is capable of deriving expressions for electronic structure methods involving unitary excitation operator upto third commutator (can be expanded). The library provides working equations in simple object form in Python thay may be helpful to meet special needs of different methods along with easy integration with other software. Section 6.2 discusses the form of expressions in UCCSD based methods. Section 6.3 discusses the data structures in the library (6.3.1), contraction using wicks theorem (6.3.2), and canonicalization of the expressions (6.3.3). Section 6.4 describes the application of the library to derive working expressions for UCC3 method.

6.2 General structure of expressions in UCC based methods

The unitary coupled-cluster wavefunction is given by an unitary parameterization of the ground state wavefunction [188–191] as

$$|\Psi_{UCC}\rangle = e^{\hat{\sigma}}|\psi_o\rangle. \tag{6.1}$$

Here the wavefunction ψ_o is ground state Hartree-Fock wavefunction and $\hat{\sigma}$ is the unitary excitation operator. Excitation operator in UCC singles and doubles (UCCSD) is given by a linear combination of single and double excitation and de-excitation operators as

$$\hat{\sigma} = \hat{\sigma}_1 + \hat{\sigma}_2, \tag{6.2}$$

where

$$\hat{\sigma}_1 = \hat{T}_1 + \hat{T}_1^\dagger, \tag{6.3}$$

$$\hat{\sigma}_2 = \hat{T}_2 + \hat{T}_2^\dagger. \tag{6.4}$$

Here, $\hat{T}_1$, $\hat{T}_1^\dagger$, $\hat{T}_2$ and $\hat{T}_2^\dagger$ are excitation and de-exctiation operators defined as

$$\hat{T}_1 = \sum_{i,a} \sigma_i^a a_a^\dagger a_i, \tag{6.5}$$

$$\hat{T}_1^\dagger = \sum_{i,a} \sigma_a^i a_i^\dagger a_a, \tag{6.6}$$

$$\hat{T}_2 = \frac{1}{4} \sum_{ij,ab} \sigma_{ij}^{ab} a_a^\dagger a_b^\dagger a_j a_i, \tag{6.7}$$

$$\hat{T}_2^\dagger = \frac{1}{4} \sum_{ij,ab} \sigma_{ab}^{ij} a_i^\dagger a_j^\dagger a_b a_a. \tag{6.8}$$

with $a, b, \ldots (i, j, \ldots)$ being unoccupied(occupied) orbitals. The operators $a^\dagger$ and a are creation and annhilation operators respectively, which create or annhilate an electron from an orbital. The unitarity of $e^{\hat{\sigma}}$ is ensured by

$$\sigma^\dagger = -\sigma, \tag{6.9}$$

therefore,

$$\sigma_i^a = -(\sigma_a^i)^*, \tag{6.10}$$

$$\sigma_{ij}^{ab} = -(\sigma_{ab}^{ij})^*. \tag{6.11}$$

The UCCSD ground state energy and amplitude equations are given by

$$E_{gr} = \langle \psi_o | \bar{H} | \psi_o \rangle, \tag{6.12}$$

$$0 = \langle \psi_l | \bar{H} | \psi_o \rangle, \tag{6.13}$$

where ψ_l's represent singly and doubly excited determinants and $\bar{H}$ is the similarity transformed Hamiltonian $\bar{H} = e^{-\hat{\sigma}} \hat{H} e^{\hat{\sigma}}$.

Complications arise in UCC theory as compared with CC theory due to the presence of both excitations and de-excitaiton operators in the exponential ansatz. Looking closely at the Backer-Campbell-Housdorff (BCH) expansion of similarity transformed Hamiltonian in the case of UCC methods, it is clear that it has a non-terminating nature due to the presence of both excitations and de-excitaitons in the $\hat{\sigma}$ operator. Note that BCH expansion of similarity transformed Hamiltonian truncates naturally at the fourth commutator in the case of CC theory. The BCH expansion of similarity transformed Hamiltonian in UCC theory is given by

$$\bar{H} = \hat{H} + [\hat{H}, \hat{\sigma}] + \frac{1}{2}[[\hat{H}, \hat{\sigma}], \hat{\sigma}] + \frac{1}{6}[[[\hat{H}, \hat{\sigma}], \hat{\sigma}], \hat{\sigma}] + \ldots, \tag{6.14}$$

$$= \hat{H} + [\hat{H}, \hat{T}] - [\hat{T}^\dagger, \hat{H}] + \frac{1}{2}\{[[\hat{H}, \hat{T}], \hat{T}] + [\hat{T}^\dagger, [\hat{T}^\dagger, \hat{H}]] - [[\hat{T}^\dagger, \hat{H}], \hat{T}] - [\hat{T}^\dagger, [\hat{H}, \hat{T}]]\} + \ldots \tag{6.15}$$

As in conventional CC methods, the commutators that contain only excitation operators (like $[[\hat{H}, \hat{T}], \hat{T}]$) or only contain de-excitation operators (like $[\hat{T}^\dagger, [\hat{T}^\dagger, \hat{H}]]$) only contribute until fourth-order of commutator. This limits the number of contributing terms that may be formed in CC methods. On the other hand, the commutators that contain both excitation and de-excitation operators (like $[\hat{T}^\dagger, [\hat{H}, \hat{T}]]$) can form non-zero contributions till infinite order of commutators and therefore the BCH series is non-truncating. This is because operators $\hat{T}^\dagger$ and $\hat{T}$ can contract between themselves and form different orders of excitation

or de-excitation operators, which will then be contracted with the Hamiltonian $\hat{H}$. Several truncation schemes have been developed to overcome the non-truncating nature of BCH expansion in UCC methods [189–194].

Another problem that arises in UCC methods is the increase in the number of contractions, and resulting unique terms that are formed for energy and amplitude terms. Let us look at a simple double commutator term in the BCH expansion Eq. 6.14 in UCC singles and double truncation,

$$[[\hat{H},\hat{\sigma}],\hat{\sigma}] =[[\hat{H},\hat{\sigma}_1],\hat{\sigma}_1] + [[\hat{H},\hat{\sigma}_2],\hat{\sigma}_2]. \tag{6.16}$$

Taking the case of a single excitation operator,

$$[[\hat{H},\hat{\sigma}_1],\hat{\sigma}_1] = [[\hat{H},\hat{T}_1],\hat{T}_1] + [\hat{T}_1^\dagger,[\hat{T}_1^\dagger,\hat{H}]] - [[\hat{T}_1^\dagger,\hat{H}],\hat{T}_1] - [\hat{T}_1^\dagger,[\hat{H},\hat{T}_1]]. \tag{6.17}$$

Notice that the only commutator that contributes in CC methods is $[[\hat{H},\hat{T}_1],\hat{T}_1]$, while the other three commutators will contribute in the case of UCC methods. This increases the complexity and number of contractions to take into consideration. Below is the expansion of all contributing operator contractions involved

$$[[\hat{H},\hat{\sigma}_1],\hat{\sigma}_1] = + \overbrace{\hat{H}\hat{T}_1\hat{T}_1} + \overbrace{\hat{T}_1^\dagger\hat{T}_1^\dagger\hat{H}} - \overbrace{\hat{T}_1^\dagger\hat{H}\hat{T}_1} - \overbrace{\hat{T}_1^\dagger\hat{H}\hat{T}_1}. \tag{6.18}$$

This increase in the number of commutators results an increase in number of intermediates and final terms in the derivation resulting in a much higher complexity compared with coupled-cluster based methods. This problem is further increased when we reach triple commutator. For instance, the contributing commutators involved in $[[[\hat{H},\hat{\sigma}_1],\hat{\sigma}_1],\hat{\sigma}_1]$ are

$$[[[\hat{H},\hat{\sigma}_1],\hat{\sigma}_1],\hat{\sigma}_1] =[[[\hat{H},\hat{T}_1],\hat{T}_1],\hat{T}_1] - [[[\hat{T}_1^\dagger,\hat{H}],\hat{T}_1],\hat{T}_1] - [[\hat{T}_1^\dagger[\hat{H},\hat{T}_1]],\hat{T}_1] \tag{6.19}$$

$$- [\hat{T}_1^\dagger,[[\hat{H},\hat{T}_1],\hat{T}_1]] + [[\hat{T}_1^\dagger,[\hat{T}_1^\dagger,\hat{H}]],\hat{T}_1] + [\hat{T}_1^\dagger,[\hat{T}_1^\dagger,[\hat{H},\hat{T}_1]]] \tag{6.20}$$

$$+ [\hat{T}_1^\dagger,[[\hat{T}_1^\dagger,\hat{H}],\hat{T}_1]] - [\hat{T}_1^\dagger,[\hat{T}_1^\dagger,[\hat{T}_1^\dagger,\hat{H}]]]. \tag{6.21}$$

The non-terminating nature of BCH expansion along with numerous possible contraction between Hamiltonian, excitation and de-excitation operators make the derivation of expressions in UCC methods time-consuming and error-prone. An automatic expression generation

and code generation may provide important support in the development of UCC based methods. In the following section, the development of an automatic expression generation library (AutoGen) is discussed.

6.3 Derivation using automatic expression generator code

From the above discussion, it is clear that the most fundamental computation is the contraction of two operators A and B as $\overbrace{AB}$ according to Wick's theorem. For expressions involving the product of more than two normal ordered operators, they can be handled sequentially. The sequential handling of more than two operators makes the library more general. Any number of contraction of operators can then theoretically be generated by this library taking two operators at a time. The following subsection discusses the data structure of operators and terms in the library. It should be noted that the discussion in this section is a highly simplified version of data structures and algorithms used in the library. This is to focus attention on the most important parts in the working of the library and keep the discussion clear.

6.3.1 Representation of operator and term in the library

An *operator* in the AutoGen library has the following form

$$\hat{V} = \frac{1}{4} \sum_{pqrs} \langle pq||rs\rangle \{a_p^\dagger a_q^\dagger a_s a_r\}. \tag{6.22}$$

The fluctuation operator $\hat{V}$ is taken as example above but all operators, such as $\hat{T}_1$, $\hat{T}_2$, etc., have the same structure in our formalism. There are three parts, a pre-factor (in this case $\frac{1}{4}$), a matrix element ($\sum_{pqrs}\langle pq||rs\rangle$) and a normal ordered string of second quantized operators ($\{a_p^\dagger a_q^\dagger a_s a_r\}$). $p, q, r, s, \ldots$ are indices used to represent general indices: they can be either occupied or unoccupied orbital indices. An operator is stored in the AutoGen library in a class named *operator* with a hierarchical data structure. Each object of the class *operator*

has three variables,

- *pre_factor*: variable that stores the numerical pre-factor of the operator,
- *matrix_element*: variable that stores the matrix element,
- *active_operator*: object that stores string of second quantized operators in normal order.

Active_operator in each operator class contains the indices of orbitals being created and annihilated as *upper* and *lower* string of indices. Here, the *upper* string stores all the creation operators, while the *lower* string stores all the annihilation operators. For instance, the *active_operator* part of the operator $\hat{V}$ may be written as $\{E^{pq}_{rs}\}$, where pq forms the upper indices and rs forms the lower indices. Notice that the brackets on $\{E^{pq}_{rs}\}$ indicate the fact that the string of second quantized operators is in normal order with respect to the reference wavefunction. All operators in the library are defined during the start of the program as objects of the class operator.

A *term* in the AutoGen library is created when two operators, an operator and a term, or two terms contract with each other. For instance, a contraction between the operators $\hat{V}$ and $\hat{T}_1$, represented as $\hat{V}\overbrace{\hat{T}_1}$, will result in the following *terms*

$$\hat{V}\overbrace{\hat{T}_1} = +\frac{1}{4}\sum_{pqrsai}\langle pq||rs\rangle\sigma^a_i\tilde{E}^{pqa}_{rsi} + \frac{1}{2}\sum_{qrsai}\langle iq||rs\rangle\sigma^a_i\tilde{E}^{qa}_{rs} + \frac{1}{2}\sum_{pqrai}\langle pq||ra\rangle\sigma^a_i\tilde{E}^{pq}_{ri} \tag{6.23}$$

$$-\sum_{qrai}\langle iq||ra\rangle\sigma^a_i\tilde{E}^{q}_{r}. \tag{6.24}$$

Each term produced in the program is stored in an object of class *term*, which has the following variables

- *pre_factor*: a variable that stores the overall numerical pre-factor of the term,
- *matrix_element*: a variable that stores the matrix element,
- *active_operator*: an object that stores the string of second quantized operators in normal order.

As an example, let us take the case of the first term in Eq. 6.23.

$$+\frac{1}{4}\sum_{pqrsai}\langle pq||rs\rangle\sigma_i^a\tilde{E}_{rsi}^{pqa}, \tag{6.25}$$

where the pre_factor is $\frac{1}{4}$, $matrix\_element$ is $\sum_{pqrsai}\langle pq||rs\rangle\sigma_i^a$ and the $active\_operator$ is $\{E_{rsi}^{pqa}\}$. The class *operator* and class *term* are similar and can be later merged for simplicity. The difference lies in the distinction between operators and terms in the current version of the library. A term may be created by the contraction of two or more operators and terms, while an operator is always pre-defined in the program. Using the definition of *operator* and *term* classes, a discussion of the creation of working expression using Wick's theorem is presented in the next subsection.

6.3.2 Contraction using Wick's theorem

Wick's theorem based contraction approach is explained in this subsection using an example of a contraction between two operators $\hat{V}$ and $\hat{T}_2$. The operators $\hat{V}$ and $\hat{T}_2$ are defined as

$$\hat{V} = \frac{1}{4}\sum_{pqrs}\langle pq||rs\rangle\{a_p^\dagger a_q^\dagger a_s a_r\}, \tag{6.26}$$

$$\hat{T}_2 = \frac{1}{4}\sum_{abij} t_{ij}^{ab}\{a_a^\dagger a_b^\dagger a_j a_i\}. \tag{6.27}$$

The contraction $\hat{V}\overbrace{\hat{T}_2}$ involves two steps: contraction of the second quantized part of the two operators using Wick's theorem, and manipulation of matrix elements to get the resulting terms. Wick's theorem-based manipulation of the two strings of normal ordered operators is carried out in the function *ewt* in the program. The two $active\_operators$ are passed on as arguments to the function. The two operators in the case of the contraction $\hat{V}\overbrace{\hat{T}_2}$ are: $\{E_{rs}^{pq}\}$ and $\{E_{ij}^{ab}\}$. The function works in the following way. First, it creates a list of indices with which the given index can contract, such as the list $\{i, j\}$ for the index p. Second, it uses the list of indices to form all possible single contractions, double contractions, and so on. The

result of procedure takes the following form

$$
\begin{aligned}
\{E^{pq}_{rs}\}\{E^{ab}_{ij}\} = &+\{E^{pqab}_{jisr}\} - \delta_{pj}\{E^{qab}_{isr}\} + \delta_{pi}\{E^{qab}_{jsr}\} + \delta_{qj}\{E^{pab}_{isr}\} \\
&- \delta_{qi}\{E^{pab}_{jsr}\} + \delta_{sa}\{E^{pqb}_{jir}\} - \delta_{sb}\{E^{pqa}_{jir}\} - \delta_{ra}\{E^{pqb}_{jis}\} \\
&+ \delta_{rb}\{E^{pqa}_{jis}\} \\
&- \delta_{pj}\delta_{qi}\{E^{ab}_{sr}\} + \delta_{pj}\delta_{sa}\{E^{qb}_{ir}\} - \delta_{pj}\delta_{sb}\{E^{qa}_{ir}\} - \delta_{pj}\delta_{ra}\{E^{qb}_{is}\} \\
&+ \delta_{pj}\delta_{rb}\{E^{qa}_{is}\} + \delta_{pi}\delta_{qj}\{E^{ab}_{sr}\} - \delta_{pi}\delta_{sa}\{E^{qb}_{jr}\} + \delta_{pi}\delta_{sb}\{E^{qa}_{jr}\} \\
&+ \delta_{pi}\delta_{ra}\{E^{qb}_{js}\} - \delta_{pi}\delta_{rb}\{E^{qa}_{js}\} - \delta_{qj}\delta_{sa}\{E^{pb}_{ir}\} + \delta_{qj}\delta_{sb}\{E^{pa}_{ir}\} \\
&+ \delta_{qj}\delta_{ra}\{E^{pb}_{is}\} - \delta_{qj}\delta_{rb}\{E^{pa}_{is}\} + \delta_{qi}\delta_{sa}\{E^{pb}_{jr}\} - \delta_{qi}\delta_{sb}\{E^{pa}_{jr}\} \\
&- \delta_{qi}\delta_{ra}\{E^{pb}_{js}\} + \delta_{qi}\delta_{rb}\{E^{pa}_{js}\} - \delta_{sa}\delta_{rb}\{E^{pq}_{ji}\} + \delta_{sb}\delta_{ra}\{E^{pq}_{ji}\} \\
&- \delta_{pj}\delta_{qi}\delta_{sa}\{E^{b}_{r}\} + \delta_{pj}\delta_{qi}\delta_{sb}\{E^{a}_{r}\} + \delta_{pj}\delta_{qi}\delta_{ra}\{E^{b}_{s}\} \\
&- \delta_{pj}\delta_{qi}\delta_{rb}\{E^{a}_{s}\} + \delta_{pj}\delta_{sa}\delta_{rb}\{E^{q}_{i}\} - \delta_{pj}\delta_{sb}\delta_{ra}\{E^{q}_{i}\} \\
&+ \delta_{pi}\delta_{qj}\delta_{sa}\{E^{b}_{r}\} - \delta_{pi}\delta_{qj}\delta_{sb}\{E^{a}_{r}\} - \delta_{pi}\delta_{qj}\delta_{ra}\{E^{b}_{s}\} \\
&+ \delta_{pi}\delta_{qj}\delta_{rb}\{E^{a}_{s}\} - \delta_{pi}\delta_{sa}\delta_{rb}\{E^{q}_{j}\} + \delta_{pi}\delta_{sb}\delta_{ra}\{E^{q}_{j}\} \\
&- \delta_{qj}\delta_{sa}\delta_{rb}\{E^{p}_{i}\} + \delta_{qj}\delta_{sb}\delta_{ra}\{E^{p}_{i}\} + \delta_{qi}\delta_{sa}\delta_{rb}\{E^{p}_{j}\} \\
&- \delta_{qi}\delta_{sb}\delta_{ra}\{E^{p}_{j}\} \\
&+ \delta_{pj}\delta_{qi}\delta_{sa}\delta_{rb} - \delta_{pj}\delta_{qi}\delta_{sb}\delta_{ra} - \delta_{pi}\delta_{qj}\delta_{sa}\delta_{rb} + \delta_{pi}\delta_{qj}\delta_{sb}\delta_{ra}
\end{aligned}
\tag{6.28}
$$

The terms in the above normal ordered form of the product of two operators contain both general indices and occupied/unoccupied indices. Notice that the last four terms in Eq. 6.28 are fully contracted terms. In the next step, the normal ordered product is merged with pre-factor and matrix elements, to form objects of class *term*. This will lead to terms of the

following form

$$
\begin{aligned}
\hat{V}\hat{T}_2 = &-\frac{1}{16}\sum_{pqrsabij}\langle pq||rs\rangle t_{ij}^{ab}E_{rsij}^{pqab} - \frac{1}{16}\sum_{qrsabij}\langle jq||rs\rangle t_{ij}^{ab}E_{rsi}^{qab} + \frac{1}{16}\sum_{qrsabij}\langle iq||rs\rangle t_{ij}^{ab}E_{rsj}^{qab} \\
&+\frac{1}{16}\sum_{prsabij}\langle pj||rs\rangle t_{ij}^{ab}E_{rsi}^{pab} - \frac{1}{16}\sum_{prsabij}\langle pi||rs\rangle t_{ij}^{ab}E_{rsj}^{pab} + \frac{1}{16}\sum_{pqrabij}\langle pq||ra\rangle t_{ij}^{ab}E_{rij}^{pqb} \\
&-\frac{1}{16}\sum_{pqrabij}\langle pq||rb\rangle t_{ij}^{ab}E_{rij}^{pqa} - \frac{1}{16}\sum_{pqsabij}\langle pq||as\rangle t_{ij}^{ab}E_{sij}^{pqb} + \frac{1}{16}\sum_{pqsabij}\langle pq||bs\rangle t_{ij}^{ab}E_{sij}^{pqa} \\
&-\frac{1}{16}\sum_{rsabij}\langle ji||rs\rangle t_{ij}^{ab}E_{rs}^{ab} + \frac{1}{16}\sum_{qrabij}\langle jq||ra\rangle t_{ij}^{ab}E_{ri}^{qb} - \frac{1}{16}\sum_{qrabij}\langle jq||rb\rangle t_{ij}^{ab}E_{ri}^{qa} \\
&-\frac{1}{16}\sum_{qsabij}\langle jq||as\rangle t_{ij}^{ab}E_{si}^{qb} + \frac{1}{16}\sum_{qsabij}\langle jq||bs\rangle t_{ij}^{ab}E_{si}^{qa} + \frac{1}{16}\sum_{rsabij}\langle ij||rs\rangle t_{ij}^{ab}E_{rs}^{ab} \\
&-\frac{1}{16}\sum_{qrabij}\langle iq||ra\rangle t_{ij}^{ab}E_{rj}^{qb} + \frac{1}{16}\sum_{qrabij}\langle iq||rb\rangle t_{ij}^{ab}E_{rj}^{qa} + \frac{1}{16}\sum_{qsabij}\langle iq||as\rangle t_{ij}^{ab}E_{sj}^{qb} \\
&-\frac{1}{16}\sum_{qsabij}\langle iq||bs\rangle t_{ij}^{ab}E_{sj}^{qa} - \frac{1}{16}\sum_{prabij}\langle pj||ra\rangle t_{ij}^{ab}E_{ri}^{pb} + \frac{1}{16}\sum_{prabij}\langle pj||rb\rangle t_{ij}^{ab}E_{ri}^{pa} \\
&+\frac{1}{16}\sum_{psabij}\langle pj||as\rangle t_{ij}^{ab}E_{si}^{pb} - \frac{1}{16}\sum_{psabij}\langle pj||bs\rangle t_{ij}^{ab}E_{si}^{pa} + \frac{1}{16}\sum_{prabij}\langle pi||ra\rangle t_{ij}^{ab}E_{rj}^{pb} \\
&-\frac{1}{16}\sum_{prabij}\langle pi||rb\rangle t_{ij}^{ab}E_{rj}^{pa} - \frac{1}{16}\sum_{psabij}\langle pi||as\rangle t_{ij}^{ab}E_{sj}^{pb} + \frac{1}{16}\sum_{psabij}\langle pi||bs\rangle t_{ij}^{ab}E_{sj}^{pa} \\
&-\frac{1}{16}\sum_{pqabij}\langle pq||ba\rangle t_{ij}^{ab}E_{ij}^{pq} + \frac{1}{16}\sum_{pqabij}\langle pq||ab\rangle t_{ij}^{ab}E_{ij}^{pq} \\
&-\frac{1}{16}\sum_{rabij}\langle ji||ra\rangle t_{ij}^{ab}E_{r}^{b} + \frac{1}{16}\sum_{rabij}\langle ji||rb\rangle t_{ij}^{ab}E_{r}^{a} + \frac{1}{16}\sum_{sabij}\langle ji||as\rangle t_{ij}^{ab}E_{s}^{b} \\
&-\frac{1}{16}\sum_{sabij}\langle ji||bs\rangle t_{ij}^{ab}E_{s}^{a} + \frac{1}{16}\sum_{qabij}\langle jq||ba\rangle t_{ij}^{ab}E_{i}^{q} - \frac{1}{16}\sum_{qabij}\langle jq||ab\rangle t_{ij}^{ab}E_{i}^{q} \\
&+\frac{1}{16}\sum_{rabij}\langle ij||ra\rangle t_{ij}^{ab}E_{r}^{b} - \frac{1}{16}\sum_{rabij}\langle ij||rb\rangle t_{ij}^{ab}E_{r}^{a} - \frac{1}{16}\sum_{sabij}\langle ij||as\rangle t_{ij}^{ab}E_{s}^{b} \\
&+\frac{1}{16}\sum_{sabij}\langle ij||bs\rangle t_{ij}^{ab}E_{s}^{a} - \frac{1}{16}\sum_{qabij}\langle iq||ba\rangle t_{ij}^{ab}E_{j}^{q} + \frac{1}{16}\sum_{qabij}\langle iq||ab\rangle t_{ij}^{ab}E_{j}^{q} \\
&-\frac{1}{16}\sum_{pabij}\langle pj||ba\rangle t_{ij}^{ab}E_{i}^{p} + \frac{1}{16}\sum_{pabij}\langle pj||ab\rangle t_{ij}^{ab}E_{i}^{p} + \frac{1}{16}\sum_{pabij}\langle pi||ba\rangle t_{ij}^{ab}E_{j}^{p} \\
&-\frac{1}{16}\sum_{pabij}\langle pi||ab\rangle t_{ij}^{ab}E_{j}^{p} \\
&+\frac{1}{16}\sum_{abij}\langle ji||ba\rangle t_{ij}^{ab} - \frac{1}{16}\sum_{abij}\langle ji||ab\rangle t_{ij}^{ab} - \frac{1}{16}\sum_{abij}\langle ij||ba\rangle t_{ij}^{ab} + \frac{1}{16}\sum_{abij}\langle ij||ab\rangle t_{ij}^{ab}
\end{aligned}
\tag{6.29}
$$

6.3.3 Canonicalization

Several terms in the above contraction procedure are equivalent due to the permutational symmetry of involved matrices. Let us take the four particle index two-electron integrals as an example to illustrate this point. Due to permutational symmetry

$$\langle ab||cd\rangle = \langle ba||cd\rangle = \langle ab||dc\rangle = \langle ba||dc\rangle. \tag{6.30}$$

Terms that arise in the $\hat{V}\hat{T}_2$ contraction that make use of these equivalent integrals may take the form

$$\langle ab||cd\rangle t_{ij}^{ab}, \qquad \langle ba||cd\rangle t_{ij}^{ab}, \qquad \langle ab||dc\rangle t_{ij}^{ab}, \qquad \langle ba||dc\rangle t_{ij}^{ab}, \tag{6.31}$$

which are all equivalent. These terms need to be added together to reach the simplest form of equations. In Eq. 6.28, among other terms, it is clear that the terms

$$+\frac{1}{16}\sum_{abij}\langle ji||ba\rangle t_{ij}^{ab} - \frac{1}{16}\sum_{abij}\langle ji||ab\rangle t_{ij}^{ab} - \frac{1}{16}\sum_{abij}\langle ij||ba\rangle t_{ij}^{ab} + \frac{1}{16}\sum_{abij}\langle ij||ab\rangle t_{ij}^{ab} \tag{6.32}$$

are all equivalent due to symmetry condition that $\langle ji||ba\rangle = \langle ji||ab\rangle = \langle ij||ba\rangle = \langle ij||ab\rangle$. This task of incorporating permutational symmetry logic to reduce the equations to reach the unique terms is a challenging task. The initial solution to this problem was provided by Kállay and Surjan [55] in their diagram logic-based approach and canonicalization was soon incorporated in TCE package [57].

In this subsection, we will discuss the algorithms in the library to reduce the expression derived by Wick's theorem to reach unique terms. The AutoGen library performs the canonicalization process after each commutator, so all the intermediates involved in the expression generation are also reduced to their simplest form. This reduces the number of terms generated after each commutator. This part of the algorithm scales as $O(N^2)$ where N is the number of terms generated by Wick's theorem. The overall performance of the algorithm is highly efficient, works well for the implemented third commutator, and can be extended to the fourth commutator and beyond.

The task of automatically recognizing equivalent terms based on permutational symmetry and merging them can be reduced to the task of comparing of two terms to determine if they are equivalent. This two-term comparison step can be repeated for each pair of terms to determine all possibilities. There are multiple methods to compare two terms. A brute force algorithm is: two terms may be said to be equivalent if they have the same form (order) of operators and orbital indices. This algorithm requires the creation of all equivalent terms of a given term based on the symmetry of matrix elements, and directly comparing the two terms based on the order of their orbital indices. For example, comparison of the terms $+\frac{1}{16}\sum_{abij}\langle ji||ba\rangle t_{ij}^{ab}$ and $-\frac{1}{16}\sum_{abij}\langle ji||ab\rangle t_{ij}^{ab}$ will first require the generation of all 16 terms that are equivalent to the first term through permutational symmetry of the matrix elements $\langle ji||ba\rangle$ and t_{ij}^{ab} in the first term. In the next step, the order of indices of each of the 16 produced terms will be directly compared to the order of indices in the second term to determine equivalence. This brute force algorithm is slow when the number of terms becomes large, as the creation of a large number of new terms with the *term* object datatype is a time-consuming process in Python.

AutoGen uses an algorithm that checks for equivalence of two terms by using logic based on diagrammatic techniques. The idea behind this comparison algorithm is that each unique term is uniquely identified by a map of the particle and hole based contraction lines between its parent operators. For instance, the terms

$$\begin{aligned} &+\frac{1}{16}\sum_{abij}\langle ji||ba\rangle t_{ij}^{ab} \\ &-\frac{1}{16}\sum_{abij}\langle ji||ab\rangle t_{ij}^{ab} \end{aligned} \tag{6.33}$$

both have two particle and two hole contraction lines between operators $\hat{V}$ and $\hat{T}_2$. Their hole and particle contraction maps are represented by the matrices

$$\begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix}, \tag{6.34}$$

where the first and second matrices represent the hole contraction line map and particle contraction line map, respectively. Note that the first column and first row represent the

operator $\hat{V}$ while the second column and second row represent the operator $\hat{T}_2$. To illustrate its working, a hypothetical extra hole contraction line between operator $\hat{V}$ and $\hat{T}_2$ will be represented by an addition in the first row second column and second row first column of the hole contraction map. The two matrices are symmetric by definition. Since both of the terms listed above will form the same contraction maps, they are determined to be equivalent by the program and may be added together. The algorithm may be broken down into three steps. First, two matrices are created storing maps of hole and particle contraction lines among parent operators, where each element of the matrix represent the number of hole or particle contraction lines that exist between the two operators represented by the row and the column. Second, the hole and particle matrices of the two terms are compared. If the hole and particle matrices of one term are equal to the hole and particle matrices of the second term, the two terms are equivalent. Third, if equivalence is established, we determine the relative sign of the two terms and add the second term to the first. The algorithm is presented in the algorithm 10. There are exceptions in this simple idea, for instance, the map of two terms where the orbital indices are permuted will also look the same to this algorithm, and need to be identified as separate. These exceptions have been identified and included in the program. They have been left out of this discussion for simplicity. The expression in Eq. 6.28 is simplified and written in canonical form as

$$\begin{aligned}\hat{V}\hat{T}_2 = &-\frac{1}{4}\sum_{qrsabij}\langle jq||rs\rangle t_{ij}^{ab}E_{rsi}^{qab} + \frac{1}{4}\sum_{pqrabij}\langle pq||ra\rangle t_{ij}^{ab}E_{rij}^{pqb} \\ &-\frac{1}{8}\sum_{rsabij}\langle ji||rs\rangle t_{ij}^{ab}E_{rs}^{ab} + 1\sum_{qrabij}\langle jq||ra\rangle t_{ij}^{ab}E_{ri}^{qb} - \frac{1}{8}\sum_{pqabij}\langle pq||ba\rangle t_{ij}^{ab}E_{ij}^{pq} \\ &-\frac{1}{2}\sum_{rabij}\langle ji||ra\rangle t_{ij}^{ab}E_{r}^{b} + \frac{1}{2}\sum_{qabij}\langle jq||ba\rangle t_{ij}^{ab}E_{i}^{q} + \frac{1}{4}\sum_{abij}\langle ji||ba\rangle t_{ij}^{ab}.\end{aligned} \tag{6.35}$$

Notice that canonicalization greatly reduces the complexity of expressions. The number of terms in the simple contraction $\hat{V}\hat{T}_2$ reduced from 48 terms to 8 terms.

Algorithm 10 algorithm for canonicalization

```
1: //allTerms ← List of all terms produced
2: procedure CANONICALIZE(allTerms)
3:     for A in allTerms do
4:         for B in terms after A do
5:             Equivalent=0
6:             HmapA → map of hole contraction lines of term A
7:             PmapA → map of particle contraction lines of term A
8:             HmapB → map of hole contraction lines of term B
9:             PmapB → map of particle contraction lines of term B
10:            if (HmapA==HmapB) and (PmapA==PmapB) then
11:                Equivalent=1
12:            else
13:                Rotate position of operators of same type and try again(e.g. $\hat{V}\hat{T}_1\hat{T}_1$ to $\hat{V}\hat{T}_1\hat{T}_1$)
14:            if Equivalent==1 then
15:                Find relative sign and merge the two terms.
```

The product of two terms, which has been the focus of the discussion so far, is used as a building block for more complex operations of deriving nested commutators. The expressions of a commutator are derived in the program by deriving the expressions of multiple operator products. For instance, the commutator

$$[\hat{V}, \hat{T}_2] = \hat{V}\hat{T}_2 - \hat{T}_2\hat{V}, \tag{6.36}$$

are derived by deriving the two operator products $\hat{V}\hat{T}_2^\dagger$, $\hat{T}_2^\dagger\hat{V}$ in the program. It should be noted that in the example given above, the contraction $\hat{T}_2\hat{V}$ will not have any terms except the fully un-contracted term. An example of nested commutator is $[[\hat{V}, \hat{T}_2], \hat{T}_2^\dagger]$, which can be expressed as a sum of operator products as

$$[[\hat{V}, \hat{T}_2], \hat{T}_2^\dagger] = \hat{V}\hat{T}_2\hat{T}_2^\dagger - \hat{T}_2\hat{V}\hat{T}_2^\dagger - \hat{T}_2^\dagger\hat{V}\hat{T}_2 + \hat{T}_2^\dagger\hat{T}_2\hat{V}. \tag{6.37}$$

This commutator is evaluated in the program using the series of single contractions, carried out one contraction at a time. To illustrate the idea, contraction $-\hat{T}_2^\dagger\hat{V}\hat{T}_2$ is carried out by

first evaluating the contraction $\hat{V}\hat{T}_2$. $\hat{T}_2^\dagger$ operator is then contracted with the result of this contraction to evaluate the expressions from the term.

The above steps for automatic expression generation for electronic structure methods have been implemented in a Python-based open-source program. Several checks are performed to check the correctness of the expressions produced. Basic check of correctness includes the production of correct expressions for CCSD energy and amplitude equations. Further checks were performed at various points of development for UCC based methods by comparing with derived expressions based on hand-drawn diagrams. The program is applied to derive working expressions for UCC based methods using terms that arise in single, double, and triple commutators in the expansion of similarity transformed Hamiltonian. The general form of terms (stored as objects of class *term*) in the program makes the program flexible. Simple functions can be developed using these *term* objects for easy post-processing on the produced terms to select, manipulate or connect with other programs. The discussion below summarizes the derived terms for UCC3 method.

6.4 Derivation of expressions for UCC based methods

Recently, a third-order polarization propagator method is developed by Liu and co-workers [188] using Bernoulli numbers as expansion coefficients in the expansion of similarity transformed Hamiltonian for the derivation of working equations (referred to as UCC3). This method is developed for ground and excited state calculations. In this section, the derivation of the working expressions of the UCC3 method using the automatic expression generator tool AutoGen is discussed.

The Hamiltonian expanded order by order in the Bernoilli numbers based expansion is given by

$$\bar{H} = \bar{H}^0 + \bar{H}^1 + \bar{H}^2 + \bar{H}^3 + \bar{H}^4, \tag{6.38}$$

where

$$\bar{H}^0 = F + V, \tag{6.39}$$

$$\bar{H}^1 = [F,\sigma] + \frac{1}{2}[V,\sigma] + \frac{1}{2}[V_R,\sigma], \tag{6.40}$$

$$\bar{H}^2 = \frac{1}{12}[[V_N,\sigma],\sigma] + \frac{1}{4}[[V,\sigma]_R,\sigma] + \frac{1}{4}[[V_R,\sigma]_R,\sigma], \tag{6.41}$$

$$\begin{aligned}\bar{H}^3 = &+ \frac{1}{24}[[[V_N,\sigma],\sigma]_R,\sigma] + \frac{1}{8}[[[V_R,\sigma]_R,\sigma]_R,\sigma] + \frac{1}{8}[[[V,\sigma]_R,\sigma]_R,\sigma] \\ &- \frac{1}{24}[[[V,\sigma]_R,\sigma],\sigma] - \frac{1}{24}[[[V_R,\sigma]_R,\sigma],\sigma],\end{aligned} \tag{6.42}$$

$$\begin{aligned}\bar{H}^4 = &\frac{1}{16}[[[[V_R,\sigma]_R,\sigma]_R,\sigma]_R,\sigma] + \frac{1}{16}[[[[V,\sigma]_R,\sigma]_R,\sigma]_R,\sigma] + \frac{1}{48}[[[[V_N,\sigma],\sigma]_R,\sigma]_R,\sigma] \\ &- \frac{1}{48}[[[[V,\sigma]_R,\sigma],\sigma]_R,\sigma] - \frac{1}{48}[[[[V_R,\sigma]_R,\sigma],\sigma]_R,\sigma] - \frac{1}{144}[[[[V_N,\sigma],\sigma]_R,\sigma],\sigma] \\ &- \frac{1}{48}[[[[V,\sigma]_R,\sigma]_R,\sigma],\sigma] - \frac{1}{48}[[[[V_R,\sigma]_R,\sigma],\sigma],\sigma] - \frac{1}{720}[[[[V_N,\sigma],\sigma],\sigma],\sigma].\end{aligned} \tag{6.43}$$

Here, the operation O_N selects the "non-diagonal" part of the operator and O_R is defined as the "rest" part of the operator ($O_R = O - O_N$). The UCC amplitude equations in this notation is written as

$$\bar{V}_N = 0. \tag{6.44}$$

In UCC-based polarisation propogator theory within the CCSD truncation scheme, the excitation energies are obtained by diagonalizing the $\bar{H}$ matrix in the singles and doubles space

$$\begin{bmatrix} \bar{H}_{SS} & \bar{H}_{SS} \\ \bar{H}_{SS} & \bar{H}_{SS} \end{bmatrix} \begin{bmatrix} C_S \\ C_D \end{bmatrix} = E \begin{bmatrix} C_S \\ C_D \end{bmatrix}. \tag{6.45}$$

The part of $\bar{H}$ relevant to UCC-based polarisation propagator theory within the singles and doubles model can be summarized as

$$\begin{aligned}\bar{H} = &E_{gr} + ((\bar{H}_{ai}\{a_a^\dagger a_i\} + \frac{1}{4}\bar{H}_{ab,ij}\{a^\dagger a^\dagger\{a_j a_i\} + h.c.) \\ &(\bar{H}_{ji}\{a_j^\dagger a_i\} + \bar{H}_{ba}\{a_b^\dagger a_a\} + \frac{1}{4}\bar{H}_{kl,ij}\{a_k^\dagger a_l^\dagger a_j a_i\} \\ &\frac{1}{4}\bar{H}_{ab,cd}\{a_a^\dagger a_b^\dagger a_d a_c\} + \bar{H}_{ia,bj}\{a_l^\dagger a_a^\dagger a_j a_b\}) \\ &+ ((\frac{1}{2}\bar{H}_{ij,ka}\{a_i^\dagger a_j^\dagger a_a a_k\} + \frac{1}{2}\bar{H}_{ci,ab}\{a_c^\dagger a_i^\dagger a_b a_a\}) + h.c.) \\ &((\frac{1}{4}\bar{H}_{ibc,abk}\{a_i^\dagger a_b^\dagger a_c^\dagger a_k a_j a_a\} + h.c.)).\end{aligned} \tag{6.46}$$

E_{gr} is the ground state UCCSD energy. The matrix elements $\bar{H}_{ai}$ and $\bar{H}_{ab,ij}$ are involved in the UCCSD amplitude equations

$$\bar{H}_{ai} = 0, \tag{6.47}$$

$$\bar{H}_{ab,ij} = 0. \tag{6.48}$$

UCC3 scheme includes the terms with the leading contributions of up to the third order in $\bar{H}_{ij}$, $\bar{H}_{ab}$, and $\bar{H}_{ia,bj}$ that contribute to $\bar{H}_{SS}$ in Eq. 6.45. UCC3 amplitude equation is solved for σ_i^a and σ_{ij}^{ab} where the matrix elements $\bar{H}_{ai}$ and $\bar{H}_{ab,ij}$ contains terms upto third order requiring derivation of double commutator between V and σ. These terms are derived using the a combination of both AutoGen library and post-processing by hand. All terms of the commutators (upto double commutator $[[V, \sigma], \sigma]$) are produced by the program. The terms are post-processed by hand to compute the action of operators $\hat{O}_R$ and $\hat{O}_N$ on the derived terms. Finally, the derived terms are compared with terms derived completely by hand using diagrammatic techniques. The derived terms for the above described parts of the

Hamiltonian are

$$\begin{aligned}\bar{H}^{UCC3}_{ai} =& \sum_{jb}\langle aj||ib\rangle\sigma^{b}_{j} + \frac{1}{2}\sum_{jb}\langle ab||ij\rangle\sigma^{b*}_{j} + \frac{1}{2}\sum_{jbc}\langle aj||cb\rangle\sigma^{cb}_{ij} - \frac{1}{2}\sum_{jkb}\langle kj||ib\rangle\sigma^{ba}_{jk} - \frac{1}{2}\sum_{jklbc}\langle al||ik\rangle\sigma^{bc*}_{jk}\sigma^{bc}_{jl} \\ & \frac{1}{2}\sum_{jkbcd}\langle ad||ic\rangle\sigma^{bd*}_{jk}\sigma^{bc}_{jk} - \sum_{jklbc}\langle bl||ji\rangle\sigma^{bc*}_{jk}\sigma^{ca}_{kl} + \sum_{jkbcd}\langle ab||dj\rangle\sigma^{bc*}_{jk} - \frac{1}{4}\sum_{jklbc}\langle bl||jk\rangle\sigma^{bc*}_{jk}\sigma^{ac}_{il} \\ & + \frac{1}{4}\sum_{jkbcd}\langle bd||jc\rangle\sigma^{bd*}_{jk}\sigma^{ac}_{ik} + \frac{1}{4}\sum_{jklbc}\langle bd||ic\rangle\sigma^{bd*}_{jk}\sigma^{ca}_{jk} - \frac{1}{4}\sum_{jkbcd}\langle al||jk\rangle\sigma^{bc*}_{jk}\sigma^{cb}_{il},\end{aligned} \tag{6.49}$$

$$\begin{aligned}\bar{H}^{UCC3}_{ab,ij} =& -P(ab)\sum_{k}\langle ka||ji\rangle\sigma^{b}_{k} + P(ij)\sum_{c}\langle ab||ic\rangle\sigma^{c}_{j} + \frac{1}{2}\sum_{kl}\langle kl||ij\rangle\sigma^{ab}_{kl} + \frac{1}{2}\sum_{cd}\langle ab||cd\rangle\sigma^{cd}_{ij} \\ & + P(ij)P(ab)\sum kc\langle ak||ic\rangle\sigma^{bc}_{jk} + P(ij)P(ab)\frac{1}{3}\sum_{klcd}\langle kl||cd\sigma^{ac}_{ik}\sigma^{bd}jl \\ & + \frac{1}{6}\sum_{klcd}\langle kl||cd\rangle\sigma^{cd}_{ij}\sigma^{ab}_{kl} - P(ab)\frac{1}{3}\sum_{klcd}\sigma^{ad}_{ij}\sigma^{cb}_{kl} - P(ij)\frac{1}{3}\sum_{klcd}\langle kl||cd\rangle\sigma^{ab}_{il}\sigma^{dc}_{jk} \\ & + P(ij)P(ab)\frac{1}{3}\sum_{klcd}\langle ad||il\rangle\sigma^{cd*}_{kl}\sigma^{bc}_{jk} + \frac{1}{12}\sum_{klcd}\langle cd||ij\rangle\sigma^{cd*}_{kl}\sigma^{ab}_{kl} + \frac{1}{12}\sum_{klcd}\langle ab||kl\rangle\sigma^{cd}_{ij}\sigma^{cd*}_{kl} \\ & - P(ab)\frac{1}{6}\sum_{klcd}\langle ad||ij\rangle\sigma^{cd*}_{kl}\sigma^{cb}_{kl} - P(ij)\frac{1}{6}\sum_{klcd}\langle ab||il\rangle\sigma^{cd*}_{kl}\sigma^{dc}_{jk} - P(ij)\frac{1}{6}\sum_{klcd}\langle cd||kj\rangle\sigma^{ab}_{il}\sigma^{cd*}_{kl} \\ & - P(ab)\frac{1}{6}\sum_{klcd}\langle bc||kl\rangle\sigma^{ad}_{ij}\sigma^{cd*}_{kl}.\end{aligned} \tag{6.50}$$

$$\bar{H}^{UCC3}_{ij} = F_{ij} + ((\sum_{ka}\langle ij||ja\rangle\sigma^{a}_{k} + \frac{1}{4}\sum_{kab}\langle ik||ab\rangle\sigma^{ab}_{jk}) + h.c.) + ((\frac{1}{2}\sum_{klabc}\langle ic||al\sigma^{ab}_{jk}\sigma^{bc*}_{kl} + \frac{1}{8}\sum_{klmab}\langle im||kl\rangle\sigma^{ab}_{jm}\sigma^{ab*}_{kl}) + h.c.) - \frac{1}{2}\sum_{klmab}\langle im||jl\sigma^{ab*}_{kl}\sigma^{ab}_{km} + \frac{1}{2}\sum_{klabc}\langle ic||jb\rangle\sigma^{ab}_{kl}\sigma^{ac*}_{kl}, \tag{6.51}$$

$$\bar{H}^{UCC3}_{ab} = F_{ab} + ((\sum_{ic}\langle ai||bc\rangle\sigma^{c}_{i} - \frac{1}{4}\sum_{ijc}\langle ij||bc\rangle\sigma^{ac}_{ij}) + h.c.) + ((-\frac{1}{2}\sum_{ijkcd}\langle kd||bj\rangle\sigma^{ca}_{ij}\sigma^{cd*}_{ij} - \frac{1}{8}\sum_{ijcdf}\langle df||cb\rangle\sigma^{ac}_{ij}\sigma^{fd*}_{ij}) + h.c.) + \frac{1}{2}\sum_{ijcdf}\langle ad||bc\rangle\sigma^{fc}_{ij}\sigma^{fd*}_{ij} - \frac{1}{2}\sum_{ijkcd}\langle ka||jb\rangle\sigma^{cd}_{ik}\sigma^{cd*}_{ij}, \tag{6.52}$$

$$\begin{aligned}\bar{H}^{UCC3}_{ia,bj} =& \langle ia||bj\rangle + ((\sum_{c}\langle ai||cb\rangle\sigma^{c}_{j} - \sum_{k}\langle ki||jb\rangle\sigma^{a}_{k} + \frac{1}{2}\sum_{kc}\langle ac||jk\rangle ac||jk\rangle\sigma^{bc*}_{ik}) + h.c.) \\ &+ ((\frac{1}{4}\sum_{klmc}\langle im||kl\rangle\sigma^{bc*}_{kl}\sigma^{ac}_{jm} + \frac{1}{4}\sum_{kcde}\langle ce||bd\rangle\sigma^{ce*}_{ik}\sigma^{ad}_{jk} - \frac{1}{2}\sum_{klcd}\langle lc||kb\rangle\sigma^{ad}_{jl}\sigma^{cd*}_{ik} - \frac{1}{2}\sum_{klcd}\langle id||kc\rangle\sigma^{ac}_{jl}\sigma^{bd*}_{kl} \\ &- \frac{1}{4}\sum^{klcd}\langle id||bj\rangle\sigma^{ca}_{kl}\sigma^{cd*}_{kl} - \frac{1}{4}\sum_{klcd}\langle ia||bl\rangle\sigma^{dc}_{jl}\sigma^{cd*}_{kl} - \sum_{klcd}\langle ia||kc\rangle\sigma^{cd}_{jl}\sigma^{bd*}_{kl} + h.c.) \\ &+ \sum_{klmc}\langle ik||lj\rangle\sigma^{ac}_{km}\sigma^{bc*}_{lm} + \sum_{klce}\langle ad||cb\rangle\sigma^{ce}_{jk}\sigma^{de*}_{ik} + \frac{1}{2]}\sum_{kldc}\langle ka||bl\rangle\sigma^{cd}_{kj}\sigma^{cd*}_{il} + \frac{1}{2}\sum_{klcd}\langle ic||dj\rangle\sigma^{cb*}_{kl}\sigma^{ad}_{kl}.\end{aligned} \tag{6.53}$$

The terms in $\bar{H}_{ci,ab}$, $\bar{H}_{jk,ia}$, $\bar{H}_{ibc,ajk}$, $\bar{H}_{ab,ci}$, $\bar{H}_{ia,jk}$, and $\bar{H}_{ajk,ibc}$ contribute to the block $\bar{H}_{DS}$ and $\bar{H}_{SD}$. Their leading contributions of first and second order are included, which are

$$\bar{H}^{UCC3}_{ci,ab} = \langle ci||ab\rangle + P(ab)\sum_{jd}\langle cd||aj\rangle\sigma^{bd*}_{ij} + \frac{1}{2}\sum_{kj}\langle ci||jk\rangle\sigma^{ab*}_{jk}, \tag{6.54}$$

$$\bar{H}^{UCC3}_{jk,ia} = \langle jk||ia\rangle + P(jk)\sum_{lb}\langle jb||il\rangle\sigma^{ab*}_{kl} + \frac{1}{2}\sum_{bc}\langle bc||ia\rangle\sigma^{bc*}_{jk}, \tag{6.55}$$

$$\bar{H}^{UCC3}_{abc,ajk} = P(jk)[-\sum_{l}\langle il||aj\rangle\sigma^{cb}_{kl}] + P(bc)[-\sum_{d}\langle ib||ad\rangle\sigma^{dc}_{jk}], \tag{6.56}$$

For $\bar{H}_{ij,kl}$ and $\bar{H}_{ab,cd}$, that contribute to $\bar{H}_{DD}$, the first-order terms are included,

$$\bar{H}^{UCC3}_{ij,kl} = \langle ij||kl\rangle, \tag{6.57}$$

$$\bar{H}^{UCC3}_{ab,cd} = \langle ab||cd\rangle, \tag{6.58}$$

Further, the terms from triple commutator are also derived using the AutoGen program for the commutator $[[[\hat{V},\sigma],\sigma],\sigma]$ to support the development of UCC-based methods correct

upto fourth order (can be found at the reference [195]). For instance, the terms from the commutator $[[[\hat{V},\sigma_1],\sigma_1],\sigma_1]$ contributing to singles amplitude ($\bar{H}_{ai}$) are given by

$$\begin{aligned}[[[\hat{V},\sigma_1],\sigma_1],\sigma_1] =& \sum_{ibck}\langle ik||cb\rangle\sigma_i^d\sigma_l^b\sigma_k^c - \sum_{iajk}\sigma_a^{i*}\langle jk||li\rangle\sigma_j^d\sigma_k^a - 1.0\sum_{iabk}\sigma_a^{i*}\langle kd||bi\rangle\sigma_l^b\sigma_k^a \\ &+ \frac{2}{3}\sum_{iack}\sigma_a^{i*}\langle kd||ci\rangle\sigma_l^a\sigma_k^c + \frac{2}{3}\sum_{iack}\sigma_a^{i*}\langle ak||lc\rangle\sigma_i^d\sigma_k^c - \sum_{iajc}\sigma_a^{i*}\langle aj||lc\rangle\sigma_j^d\sigma_i^c \\ &- \sum_{iabc}\sigma_a^{i*}\langle ad||cb\rangle\sigma_l^b\sigma_i^c - \sum_{iajc}\sigma_a^{i*}\langle aj||ci\rangle\sigma_j^d\sigma_l^c - \frac{2}{3}\sum_{jaic}\sigma_a^{j*}\langle id||lc\rangle\sigma_i^a\sigma_j^c \\ &\frac{1}{3}\sum_{ciaj}\sigma_c^{j*}\langle cd||li\rangle\sigma_a^{i*}\sigma_j^a - \frac{1}{6}\sum_{kcia}\sigma_c^{k*}\langle cd||ki\rangle\sigma_a^{i*}\sigma_l^a - \frac{1}{6}\sum_{kcia}\sigma_c^{k*}\langle ac||lk\rangle\sigma_a^{i*}\sigma_i^d. \end{aligned} \tag{6.59}$$

6.5 Summary

An automatic expression generation library, named AutoGen, has been presented in the above chapter; along with the derivation of working expressions for the third-order UCC-based polarisation propagator method (UCC3). AutoGen provides a powerful initial framework for rapid development of quantum chemistry programs, and it is especially suited for UCC-based methods. Currently, it is capable of deriving tensor contraction expressions up to the third-order commutator of UCC-based methods, which can be expanded. The library is based on an adaptable object-oriented framework where the resulting *term* objects can be extracted and used to develop simple functions as add-ons to include special needs of the method or use in other software packages. AutoGen library provides a powerful initial framework for rapid development and testing of UCC-based methods. Further, the program is freely available and is open-source with the hope that it may be useful for researchers, educators, and students in the field of quantum chemistry.

Chapter 7

Conclusion and future work

This dissertation contributes to two topics in electronic structure method development: relativistic spin-orbit coupled-cluster methods and automatic implementation techniques. In the first part, efficient implementations of spin-orbit coupled-cluster methods have been developed using sparse AO matrix-based algorithms, leveraging spin-free nature of AO two-electron integrals and sparsity in AO integral matrix. The second part deals with the development of a Python library (AutoGen) to efficiently derive working equations of electronic structure methods using Wick's theorem, and its application to derive working equations of UCC3 method.

In the first part of this dissertation, in chapters 3 to 5, an efficient implementation of SO-CC methods has been developed using sparse AO matrix-based algorithms. SO-CC methods have been limited in their application to small molecules, compared with non-relativistic CC methods, due to higher requirements related to computational wall time and storage space [102, 144]. The implementation presented in this dissertation eliminates a major bottleneck in SO-CC methods related to their storage requirements. The sparse AO matrix-based algorithms reduce the storage required in SO-CC methods by an order of magnitude, extending the applicability of SO-CC methods to larger heavy element containing molecules. The storage required by the most storage-intensive molecular orbital (MO) integral matrix $\langle ab||cd\rangle$ is avoided in SO-CCSD and SO-EOM-CCSD methods using sparse AO matrix-based algorithms in chapter 4, and storage required by MO integral matrix $\langle ab||ci\rangle$ is avoided in

SO-CCSD(T) method using sparse AO matrix-based algorithms in chapter 5. Sparsity in AO integral matrix has been efficiently used in these algorithms to further reduce storage space requirements and formal floating point operation count. It is important to note that no approximations are introduced in these developments. The implementation is efficiently parallelized using shared memory (OpenMP) based parallelization technique, where efficient load-balancing techniques have been used for sparse matrix-based multiplications in the algorithm. The implementation has been developed in the quantum chemistry package CFOUR, and will be available for public use in the future release. The implementation extends the applicability of the methods to larger heavy-element-containing molecules. For example, the storage requirements for SO-CCSD(T) calculation of UF_6 molecule is reduced from 5 TB in MO-based implementation to about 1 TB, thereby bringing such calculations within reach of available computational resources.

A near-term plan is to use the implementation of SO-CCSD(T) method developed to study the ground state properties of actinide containing molecules such as UF_6. Further development is also planned to formulate and implement sparse AO matrix-based algorithms to avoid $\langle ab||ci\rangle$-type integral matrix in SO-EOM-CCSD method. This implementation will reduce storage requirements further in the SO-EOM-CCSD method and may allow the accurate study of excitation energies of larger molecules containing heavy elements. Further, since high computational wall time and storage requirements are primary bottlenecks related to the applicability of SO-CC methods, the development of pair natural orbital (PNO)-SO-CC based methods could drastically extend the capacity of SO-CC methods to much larger molecules. Local PNO based CC methods have proved successful for chemical studies of large molecules in non-relativistic domain [196–198]. The time required for integral transformation step in PNO-SO-CC methods, which is a time-consuming step, can further be reduced by developing resolution of identity (RI) [147–149] or Cholesky decomposition (CD)-based [152–156] implementation. These developments may reduce the computational requirements and allow the study of much larger molecules containing heavy elements with

reasonable accuracy. Reducing storage requirements of SO-CC methods also facilitates the implementation of SO-CC methods using a hybrid MPI (Message Passing Interface)-OpenMP approach of parallelization, which employ both shared memory intra-note and distributed memory multinode parallelization. Such highly-parallelized implementations have been extensively developed for non-relativistic CC methods [142, 162, 199–204]. This may extend the reach of accurate SO-CC methods to larger molecules through reducing the wall time for calculations by distributing the computation to a number of compute nodes, accessing hundreds of processors. Such an implementation may be useful to provide accurate benchmark calculations for larger heavy-element-containing molecules, to assess the impact of any approximations introduced to reduce the complexity of SO-CC methods [162].

The second part of this dissertation, in chapter 6, deals with the development of an automatic expression generation library (AutoGen). This tool uses Wick's theorem to derive expressions for electronic structure methods. Working expressions for UCC3 method are presented that are derived using the AutoGen library. Autogen is a python-based, open-source library that is highly flexible for further development based on unique requirements of the methods and can be combined with various chemistry software for additional functionality due to its object-oriented structure. AutoGen library provides a powerful initial framework for the rapid development and testing of quantum chemistry methods, and it is especially well suited for UCC-based methods.

There has been renewed interest in the development of UCC-based electronic structure methods recently [188, 189, 205–212] due to their applications in quantum computing (please refer to [213] for a recent review), their guaranteed real eigenvalues because of a Hermitian Hamiltonian [214], and their potential applications to treat same-symmetry conical intersection [214–217]. Further development of AutoGen library is planned to allow automatic implementation of working equations of UCC-based methods in a quantum chemistry package. This can be carried out in a number of ways, including integration with an existing code development tool (such as TCE [57]), translating the tensor contraction expressions into

working code using NumPy einsum [218] (for a recent example of this strategy please refer to paper by Rubin and DePrince [132]), or developing new functions that translate tensor contraction expressions into efficiently written code. A potential application of such a library is to facilitate testing of various truncation schemes of UCC [189–194] for an efficient initialization of the cluster amplitudes for Variational Quantum Eigensolver (VQE) algorithms [205]. Such a study may be useful to find optimal initial wavefunction amplitudes for VQE hybrid approach to quantum computing for further optimization. This may help avoid the potential difficulties posed by a random initialization of amplitudes (Barren plateaus) [219] and potentially reduce the number of iterations needed for the VQE optimization. Another planned direction in the development of the AutoGen library is to add the functionality of automatic derivation of working expressions using Kutzelnigg and Mukherjee [180] type generalized normal ordering. The AutoGen library is currently already capable of doing spin-free normal ordering based on Kutzelnigg and Mukherjee type generalized normal ordering for second quantized operators [180, 220–222]. Extention of the library to derive working expressions and further code generation using Kutzelnigg and Mukherjee type generalized normal ordering may support the study of the multi-reference problem in chemistry [39–46].

References

[1]C. D. Sherrill, D. E. Manolopoulos, T. J. Martínez, and A. Michaelides, "Electronic structure software," J. Chem. Phys. **153** (2020).

[2]"Wikipedia contributors, list of quantum chemistry and solid-state physics software—wikipedia, the free encyclopedia, 2020; accessed april 2021.,"

[3]J. F. Stanton, J. Gauss, L. Cheng, M. E. Harding, D. A. Matthews, and P. G. Szalay, *CFOUR, Coupled-Cluster techniques for Computational Chemistry, a quantum-chemical program package*, With contributions from A.A. Auer, R.J. Bartlett, U. Benedikt, C. Berger, D.E. Bernholdt, S. Blaschke, Y. J. Bomble, S. Burger, O. Christiansen, D. Datta, F. Engel, R. Faber, J. Greiner, M. Heckert, O. Heun, M. Hilgenberg, C. Huber, T.-C. Jagau, D. Jonsson, J. Jusélius, T. Kirsch, K. Klein, G.M. KopperW.J. Lauderdale, F. Lipparini, T. Metzroth, L.A. Mück, D.P. O'Neill, T. Nottoli, D.R. Price, E. Prochnow, C. Puzzarini, K. Ruud, F. Schiffmann, W. Schwalbach, C. Simmons, S. Stopkowicz, A. Tajti, J. Vázquez, F. Wang, J.D. Watts and the integral packages MOLECULE (J. Almlöf and P.R. Taylor), PROPS (P.R. Taylor), ABACUS (T. Helgaker, H.J. Aa. Jensen, P. Jørgensen, and J. Olsen), and ECP routines by A. V. Mitin and C. van Wüllen. For the current version, see http://www.cfour.de.

[4]D. A. Matthews, L. Cheng, M. E. Harding, F. Lipparini, S. Stopkowicz, T. C. Jagau, P. G. Szalay, J. Gauss, and J. F. Stanton, "Coupled-cluster techniques for computational chemistry: The CFOUR program package," J. Chem. Phys. **152**, 214108 (2020).

[5]M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox, *Gaussian~16 Revision C.01*, Gaussian Inc. Wallingford CT, 2016.

[6]F. Neese, "The ORCA program system," Wiley Interdiscip. Rev. Comput. Mol. Sci. **2**, 73–78 (2012).

[7]G. M. J. Barca, C. Bertoni, L. Carrington, D. Datta, N. De Silva, J. E. Deustua, D. G. Fedorov, J. R. Gour, A. O. Gunina, E. Guidez, T. Harville, S. Irle, J. Ivanic, K. Kowalski, S. S. Leang, H. Li, W. Li, J. J. Lutz, I. Magoulas, J. Mato, V. Mironov, H. Nakata, B. Q. Pham, P. Piecuch,

D. Poole, S. R. Pruitt, A. P. Rendell, L. B. Roskop, K. Ruedenberg, T. Sattasathuchana, M. W. Schmidt, J. Shen, L. Slipchenko, M. Sosonkina, V. Sundriyal, A. Tiwari, J. L. Galvez Vallejo, B. Westheimer, M. Wloch, P. Xu, F. Zahariev, and M. S. Gordon, "Recent developments in the general atomic and molecular electronic structure system," en, J. Chem. Phys. **152**, 154102 (2020).

[8]G. Karlström, R. Lindh, P. Å. Malmqvist, B. O. Roos, U. Ryde, V. Veryazov, P. O. Widmark, M. Cossi, B. Schimmelpfennig, P. Neogrady, and L. Seijo, "MOLCAS: A program package for computational chemistry," Comput. Mater. Sci. **28**, 222–239 (2003).

[9]C. Peng, C. A. Lewis, X. Wang, M. C. Clement, K. Pierce, V. Rishi, F. Pavošević, S. Slattery, J. Zhang, N. Teke, A. Kumar, C. Masteran, A. Asadchev, J. A. Calvin, and E. F. Valeev, "Massively Parallel Quantum Chemistry: A high-performance research platform for electronic structure," J. Chem. Phys. **153** (2020).

[10]T. Saue, R. Bast, A. S. P. Gomes, H. J. A. Jensen, L. Visscher, I. A. Aucar, R. Di Remigio, K. G. Dyall, E. Eliav, E. Fasshauer, T. Fleig, L. Halbert, E. D. Hedegård, B. Helmich-Paris, M. Iliaš, C. R. Jacob, S. Knecht, J. K. Laerdahl, M. L. Vidal, M. K. Nayak, M. Olejniczak, J. M. H. Olsen, M. Pernpointner, B. Senjean, A. Shee, A. Sunaga, and J. N. van Stralen, "The DIRAC code for relativistic molecular calculations," J. Chem. Phys. **152**, 204104 (2020).

[11]J. F. Stanton, J. Gauss, J. D. Watts, W. J. Lauderdale, and R. J. Bartlett, "The ACES II program system," J. Quantum Chem. **44**, 879–894 (1992).

[12]Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters, and G. K. L. Chan, "PySCF: the Python-based simulations of chemistry framework," Wiley Interdiscip. Rev. Comput. Mol. Sci. **8** (2018).

[13]J. M. Turney, A. C. Simmonett, R. M. Parrish, E. G. Hohenstein, F. A. Evangelista, J. T. Fermann, B. J. Mintz, L. A. Burns, J. J. Wilke, M. L. Abrams, N. J. Russ, M. L. Leininger, C. L. Janssen, E. T. Seidl, W. D. Allen, H. F. Schaefer, R. A. King, E. F. Valeev, C. D. Sherrill, and T. D. Crawford, "Psi4: An open-source ab initio electronic structure program," Wiley Interdiscip. Rev. Comput. Mol. Sci. **2**, 556–565 (2012).

[14]M. Valiev, E. J. Bylaska, N. Govind, K. Kowalski, T. P. Straatsma, H. J. Van Dam, D. Wang, J. Nieplocha, E. Apra, T. L. Windus, and W. A. De Jong, "NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations," Comput. Phys. Comm. **181**, 1477–1489 (2010).

[15]H. J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, and M. Schütz, "Molpro: A general-purpose quantum chemistry program package," Wiley Interdiscip. Rev. Comput. Mol. Sci. **2**, 242–253 (2012).

[16]D. A. Case, T. A. Darden, T. E. Cheatham, C. L. Simmerling, J. Wang, R. E. Duke, R. Luo, M. Crowley, R. C. Walker, W. Zhang, et al., *Amber 10*, tech. rep. (University of California, 2008).

[17]D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. Berendsen, "Gromacs: fast, flexible, and free," J. Comput. Chem. **26**, 1701–1718 (2005).

[18]B. R. Brooks, C. L. Brooks III, A. D. Mackerell Jr, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, et al., "Charmm: the biomolecular simulation program," J. Comput. Chem. **30**, 1545–1614 (2009).

[19]W. Humphrey, A. Dalke, and K. Schulten, "Vmd: visual molecular dynamics," J. Mol. Graph. **14**, 33–38 (1996).

[20]"Https://portal.nersc.gov/project/mpccc/baustin/nersc_2014_workload_analysis_v1.1.pdf," accessed April 2021. (2015).

[21]J. A. Pople, "Nobel lecture: Quantum chemical models," Rev. Mod. Phys. **71**, 1267–1274 (1999).

[22]R. J. Bartlett and M. Musiał, "Coupled-cluster theory in quantum chemistry," Rev. Mod. Phys. **79**, 291 (2007).

[23]K. Burke, "Perspective on density functional theory," J. Chem. Phys. **136**, 150901 (2012).

[24]K. G. Dyall and K. Fægri Jr, *Introduction to relativistic quantum chemistry* (Oxford University Press, 2007).

[25]M. Reiher and A. Wolf, *Relativistic quantum chemistry: the fundamental theory of molecular science* (John Wiley & Sons, 2014).

[26]W. Kutzelnigg, "Solved and unsolved problems in relativistic quantum chemistry," Chem. Phys. **395**, 16–34 (2012).

[27]W. Liu, *Handbook of relativistic quantum chemistry* (Springer Berlin Heidelberg Berlin, Heidelberg, 2017).

[28]P. Pyykkö, "Relativistic quantum chemistry," in *Adv. quantum chem.* Vol. 11 (Elsevier, 1978), pp. 353–409.

[29]P. Pyykko and J. P. Desclaux, "Relativity and the periodic system of elements," Acc. Chem. Res. **12**, 276–281 (1979).

[30]P. Pyykkö, "Relativistic Effects in Structural Chemistry," Chem. Rev. **88**, 563–594 (1988).

[31]K. S. Pitzer, "Relativistic effects on chemical properties," Acc. Chem. Res. **12**, 271–276 (1979).

[32]P. A. Christiansen, W. C. Ermler, and K. S. Pitzer, "Relativistic effects in chemical systems," Annu. Rev. Phys. Chem. **36**, 407–432 (1985).

[33]T. Fleig, "Invited review: Relativistic wave-function based electron correlation methods," Chem. Phys. **395**, 2–15 (2012).

[34]J. Liu and L. Cheng, "Relativistic coupled-cluster and equation-of-motion coupled-cluster methods," Wiley Interdiscip. Rev. Comput. Mol. Sci., e1536 (2021).

[35]A. Tajti, P. G. Szalay, A. G. Császár, M. Kállay, J. Gauss, E. F. Valeev, B. A. Flowers, J. Vázquez, and J. F. Stanton, "Heat: high accuracy extrapolated ab initio thermochemistry," J. Chem. Phys. **121**, 11599–11613 (2004).

[36]A. Karton, E. Rabinovich, J. M. Martin, and B. Ruscic, "W4 theory for computational thermochemistry: in pursuit of confident sub-kj/mol predictions," J. Chem. Phys. **125**, 144108 (2006).

[37]K. A. Peterson, D. Feller, and D. A. Dixon, "Chemical accuracy in ab initio thermochemistry and spectroscopy: current strategies and future challenges," Theor. Chem. Acc. **131**, 1–20 (2012).

[38]W. Jiang, N. J. DeYonker, J. J. Determan, and A. K. Wilson, "Toward accurate theoretical thermochemistry of first row transition metal complexes," J. Phys. Chem. A **116**, 870–885 (2012).

[39]C. D. Sherrill, "Frontiers in electronic structure theory," J. Chem. Phys. **132** (2010).

[40]C. D. Sherrill, "Bond breaking in quantum chemistry," Annu. Rep. Comput. Chem. **1**, 45–56 (2005).

[41]M. W. Schmidt and M. S. Gordon, "The construction and interpretation of mcscf wavefunctions," Annu. Rev. Phys. Chem. **49**, 233–266 (1998).

[42]F. Neese, D. G. Liakos, and S. Ye, "Correlated wavefunction methods in bioinorganic chemistry," J. Bio. Inorg. Chem. **16**, 821–829 (2011).

[43]F. A. Evangelista, "Perspective: multireference coupled cluster theories of dynamical electron correlation," J. Chem. Phys. **149**, 030901 (2018).

[44]U. S. Mahapatra, B. Datta, B. Bandyopadhyay, and D. Mukherjee, "State-specific multi-reference coupled cluster formulations: two paradigms," Adv. Quantum Chem. **30**, 163–193 (1998).

[45]D. I. Lyakh, M. Musiał, V. F. Lotrich, and R. J. Bartlett, "Multireference nature of chemistry: the coupled-cluster view," Chem. Rev. **112**, 182–243 (2012).

[46]T. Shiozaki and H.-J. Werner, "Multireference explicitly correlated f12 theories," Mol. Phys. **111**, 607–630 (2013).

[47]P. Pulay, "Localizability of dynamic electron correlation," Chem. Phys. Lett. **100**, 151–154 (1983).

[48]S. Saebo and P. Pulay, "Local treatment of electron correlation," Annu. Rev. Phys. Chem. **44**, 213–236 (1993).

[49]C. Hampel and H. J. Werner, "Local treatment of electron correlation in coupled cluster theory," J. Chem. Phys. **104**, 6286–6297 (1996).

[50]M. Schütz and H. J. Werner, "Local perturbative triples correction (T) with linear cost scaling," Chem. Phys. Lett. **318**, 370–378 (2000).

[51]M. Saitow, U. Becker, C. Riplinger, E. F. Valeev, and F. Neese, "A new near-linear scaling, efficient and accurate, open-shell domain-based local pair natural orbital coupled cluster singles and doubles theory," J. Chem. Phys. **146**, 164105 (2017).

[52]Y. Guo, C. Riplinger, U. Becker, D. G. Liakos, Y. Minenkov, L. Cavallo, and F. Neese, "Communication: an improved linear scaling perturbative triples correction for the domain based local pair-natural orbital based singles and doubles coupled cluster method [dlpno-ccsd (t)]," J. Chem. Phys. **148**, 011101 (2018).

[53]M. Hanauer and A. Köhn, "Pilot applications of internally contracted multireference coupled cluster theory, and how to choose the cluster operator properly," J. Chem. Phys. **134** (2011).

[54]E. Neuscamman, T. Yanai, and G. K. L. Chan, "A review of canonical transformation theory," Int. Rev. Phys. Chem. **29**, 231–271 (2010).

[55]M. Kállay and P. R. Surján, "Higher excitations in coupled-cluster theory," J. Chem. Phys. **115**, 2945–2954 (2001).

[56]C. Song, L. P. Wang, and T. J. Martínez, "Automated Code Engine for Graphical Processing Units: Application to the Effective Core Potential Integrals and Gradients," J. Chem. Theory Comput. **12**, 92–106 (2016).

[57]S. Hirata, "Symbolic algebra in quantum chemistry," Theor. Chem. Acc. **116**, 2–17 (2006).

[58]M. Krupička, K. Sivalingam, L. Huntington, A. A. Auer, and F. Neese, "A toolchain for the automatic generation of computer codes for correlated wavefunction calculations," J. Comput. Chem. **38**, 1853–1868 (2017).

[59]A. Hartono, A. Sibiryakov, M. Nooijen, G. Baumgartner, D. E. Bernholdt, S. Hirata, C.-C. Lam, R. M. Pitzer, J. Ramanujam, and P. Sadayappan, "Automated operation minimization of tensor contraction expressions in electronic structure calculations," in International conference on computational science (Springer, 2005), pp. 155–164.

[60]A. Krylov, T. L. Windus, T. Barnes, E. Marin-Rimoldi, J. A. Nash, B. Pritchard, D. G. Smith, D. Altarawy, P. Saxe, C. Clementi, T. D. Crawford, R. J. Harrison, S. Jha, V. S. Pande, and T. Head-Gordon, "Perspective: Computational chemistry software and its advancement as illustrated through three grand challenge cases for molecular science," J. Chem. Phys. **149** (2018).

[61]T. Crawford, R. Harrison, A. I. Krylov, T. Windus, E. Carter, E. Chow, E. Deumens, M. Gordon, M. Head-Gordon, T. Martinez, et al., "Sustainable software for computational chemistry and materials modeling," (2012).

[62]P. A. M. Dirac, "Quantum mechanics of many-electron systems," Proc. R. Soc. Lond. Series A, Containing Papers of a Mathematical and Physical Character **123**, 714–733 (1929).

[63]D. R. Yarkony, "International Reviews in Physical Chemistry Spin-forbidden chemistry within the Breit-Pauli approximation Spin-forbidden chemistry within the Breit-Pauli approximation," Int. Rev. Phys. Chem. **11**, 195–242 (1992).

[64]J. Autschbach, "Perspective: Relativistic effects," J. Chem. Phys. **136** (2012).

[65]L. L. Foldy and S. A. Wouthuysen, "On the dirac theory of spin 1/2 particles and its non-relativistic limit," Phys. Rev. **78**, 29 (1950).

[66]B. A. Hess, "Relativistic electronic-structure calculations employing a two-component no-pair formalism with external-field projection operators," Phys. Rev. A **33**, 3742 (1986).

[67]K. G. Dyall, "Interfacing relativistic and nonrelativistic methods. I. Normalized elimination of the small component in the modified Dirac equation," J. Chem. Phys. **106**, 9618–9626 (1997).

[68]W. Kutzelnigg and W. Liu, "Quasirelativistic theory equivalent to fully relativistic theory," J. Chem. Phys. **123** (2005).

[69]W. Liu and D. Peng, "Exact two-component Hamiltonians revisited," J. Chem. Phys. **131**, 1–5 (2009).

[70]M. Reiher, "Douglas–kroll–hess theory: a relativistic electrons-only theory for chemistry," Theor. Chem. Acc. **116**, 241–252 (2006).

[71]D. Peng and M. Reiher, "Exact decoupling of the relativistic fock operator," Theor. Chem. Acc. **131**, 1–20 (2012).

[72]E. v. Lenthe, E.-J. Baerends, and J. G. Snijders, "Relativistic regular two-component hamiltonians," J. Chem. Phys. **99**, 4597–4610 (1993).

[73]M. Barysz and A. J. Sadlej, "Infinite-order two-component theory for relativistic quantum chemistry," J. Chem. Phys. **116**, 2696–2704 (2002).

[74]T. Saue, "Relativistic Hamiltonians for Chemistry: A Primer," ChemPhysChem **12**, 3077–3094 (2011).

[75]M. Iliaš and T. Saue, "An infinite-order two-component relativistic Hamiltonian by a simple one-step transformation," J. Chem. Phys. **126** (2007).

[76]L. L. Foldy, "On the Dirac Theory of Spin 1 / 2 Particles and Its Non-Relativistic," Phys. Rev. **78**, 29–36 (1990).

[77]C. Chang, M. Pelissier, and P. Durand, "Regular two-component pauli-like effective hamiltonians in dirac theory," Physica Scripta **34**, 394 (1986).

[78]M. Douglas and N. M. Kroll, "Quantum electrodynamical corrections to the fine structure of helium," Ann. Phys. **82**, 89–155 (1974).

[79]L. Visscher and K. G. Dyall, "Relativistic and correlation effects on molecular properties. I. The dihalogens F2, Cl2, Br2, I2, and At2," J. Chem. Phys. **104**, 9040–9046 (1996).

[80]L. Visscher, T. J. Lee, and K. G. Dyall, "Formulation and implementation of a relativistic unrestricted coupled-cluster method including noniterative connected triples," J. Chem. Phys. **105**, 8769–8776 (1996).

[81]I. Shavitt and R. J. Bartlett, *Many-body methods in chemistry and physics: mbpt and coupled-cluster theory* (Cambridge university press, 2009).

[82]T. D. Crawford and H. F. Schaefer, "An introduction to coupled cluster theory for computational chemists," Rev. Comput. Chem. **14**, 33–136 (2000).

[83]K. Raghavachari, G. W. Trucks, J. A. Pople, and M. Head-Gordon, "A fifth-order perturbation comparison of electron correlation theories," Chem. Phys. Lett. **157**, 479–483 (1989).

[84]J. F. Stanton, "Why CCSD(T) works: a different perspective," Chem. Phys. Lett. **281**, 130–134 (1997).

[85]U. Kaldor and E. Eliav, "High-accuracy calculations for heavy and super-heavy elements," Adv. Quantum Chem. **31**, 313–336 (1998).

[86]T. Leininger, A. Nicklass, H. Stoll, M. Dolg, and P. Schwerdtfeger, "The accuracy of the pseudopotential approximation. ii. a comparison of various core sizes for indium pseudopotentials in calculations for spectroscopic constants of inh, inf, and incl," J. Chem. Phys. **105**, 1052–1059 (1996).

[87]A. Wolf, M. Reiher, and B. A. Hess, "The generalized douglas–kroll transformation," J. Chem. Phys. **117**, 9215–9226 (2002).

[88]U. Kaldor and B. A. Heß, "Relativistic all-electron coupled-cluster calculations on the gold atom and gold hydride in the framework of the douglas-kroll transformation," Chem. Phys. Lett. **230**, 1–7 (1994).

[89]K. G. Dyall, "An exact separation of the spin-free and spin-dependent terms of the Dirac-Coulomb-Breit Hamiltonian," J. Chem. Phys. **100**, 2118–2127 (1994).

[90]L. Cheng and J. Gauss, "Analytical evaluation of first-order electrical properties based on the spin-free dirac-coulomb hamiltonian," J. Chem. Phys. **134**, 244112 (2011).

[91]K. A. Peterson, B. C. Shepler, D. Figgen, and H. Stoll, "On the spectroscopic and thermochemical properties of clo, bro, io, and their anions," J. Phys. Chem. A **110**, 13877–13883 (2006).

[92]N. J. DeYonker, K. A. Peterson, G. Steyl, A. K. Wilson, and T. R. Cundari, "Quantitative computational thermochemistry of transition metal species," J. Phys. Chem. A **111**, 11269–11277 (2007).

[93]D. Feller, K. A. Peterson, and D. A. Dixon, "Further benchmarks of a composite, convergent, statistically calibrated coupled-cluster-based approach for thermochemical and spectroscopic studies," Mol. Phys. **110**, 2381–2399 (2012).

[94]T. Fleig and L. Visscher, "Large-scale electron correlation calculations in the framework of the spin-free dirac formalism: the au2 molecule revisited," Chem. Phys. **311**, 113–120 (2005).

[95]L. Cheng, J. Gauss, B. Ruscic, P. B. Armentrout, and J. F. Stanton, "Bond dissociation energies for diatomic molecules containing 3d transition metals: benchmark scalar-relativistic coupled-cluster calculations for 20 molecules," J. Chem. Theory Comput. **13**, 1044–1056 (2017).

[96]L. Cheng and J. Gauss, "Perturbative treatment of spin-orbit coupling within spin-free exact two-component theory," J. Chem. Phys. **141**, 164107 (2014).

[97]L. Cheng, F. Wang, J. F. Stanton, and J. Gauss, "Perturbative treatment of spin-orbit-coupling within spin-free exact two-component theory using equation-of-motion coupled-cluster methods," J. Chem. Phys. **148**, 044108 (2018).

[98]L. Visscher, K. G. Dyall, and T. J. Lee, "Kramers-Restricted Closed-Shell CCSD Theory," Int. J. Quant. Chem.: Quant. Chem. Symp. **419**, 411–419 (1995).

[99]L. Visscher, T. J. Lee, and K. G. Dyall, "Formulation and implementation of a relativistic unrestricted coupled-cluster method including noniterative connected triples," J. Chem. Phys. **105**, 8769–8776 (1996).

[100]E. Eliav, U. Kaldor, and Y. Ishikawa, "Open-shell relativistic coupled-cluster method with dirac-fock-breit wave functions: energies of the gold atom and its cation," Phys. Rev. A **49**, 1724 (1994).

[101]J. Sikkema, L. Visscher, T. Saue, and M. Iliaš, "The molecular mean-field approach for correlated relativistic calculations," J. Chem. Phys. **131** (2009).

[102]M. Iliaš, V. Kellö, L. Visscher, and B. Schimmelpfennig, "Inclusion of mean-field spin-orbit effects based on all-electron two-component spinors: Pilot calculations on atomic and molecular properties," J. Chem. Phys. **115**, 9667–9674 (2001).

[103]S. Y. Lee and Y. S. Lee, "Kramers' restricted hartree—fock method for polyatomic molecules using ab initio relativistic effective core potentials with spin—orbit operators," J. Comput. Chem. **13**, 595–601 (1992).

[104]H. S. Lee, W. K. Cho, Y. J. Choi, and Y. S. Lee, "Spin-orbit effects for the diatomic molecules containing halogen elements studied with relativistic effective core potentials: HX, X2 (X = Cl, Br and I) and IZ (Z = F, Cl and Br) molecules," Chem. Phys. **311**, 121–127 (2005).

[105]L. Belpassi, I. Infante, F. Tarantelli, and L. Visscher, "The chemical bond between Au(I) and the noble gases. Comparative study of NgAuF and NgAu+ (Ng = Ar, Kr, Xe) by density functional and coupled cluster methods," J. Am. Chem. Soc. **130**, 1048–1060 (2008).

[106]M. Kállay, H. S. Nataraj, B. K. Sahoo, B. P. Das, and L. Visscher, "Relativistic general-order coupled-cluster method for high-precision calculations: Application to the Al${}
{+}$ atomic clock," Phys. Rev. A **83**, 30503 (2011).

[107]R. L. Haiduke, A. B. Da Silva, and L. Visscher, "The nuclear electric quadrupole moment of antimony from the molecular method," J. Chem. Phys. **125** (2006).

[108]L. F. Pašteka, E. Eliav, A. Borschevsky, U. Kaldor, and P. Schwerdtfeger, "Relativistic Coupled Cluster Calculations with Variational Quantum Electrodynamics Resolve the Discrepancy between Experiment and Theory Concerning the Electron Affinity and Ionization Potential of Gold," Phys. Rev. Lett. **118**, 23002 (2017).

[109]E. Eliav, U. Kaldor, and B. A. Hess, "The relativistic Fock-space coupled-cluster method for molecules: CdH and its ions," J. Chem. Phys. **108**, 3409–3415 (1998).

[110]N. S. Mosyagin, A. V. Titov, E. Eliav, and U. Kaldor, "Generalized relativistic effective core potential and relativistic coupled cluster calculation of the spectroscopic constants for the HgH molecule and its cation," J. Chem. Phys. **115**, 2007–2013 (2001).

[111]T. A. Isaev, A. N. Petrov, N. S. Mosyagin, A. V. Titov, E. Eliav, and U. Kaldor, "In search of the electron dipole moment: Ab initio calculations on 207PbO excited states," Phys. Rev. A - Atomic, Molecular, and Optical Physics **69**, 22–25 (2004).

[112]F. Wang, J. Gauss, and C. Van Wüllen, "Closed-shell coupled-cluster theory with spin-orbit coupling," J. Chem. Phys. **129** (2008).

[113]Z. Wang, Z. Tu, and F. Wang, "Equation-of-motion coupled-cluster theory for excitation energies of closed-shell systems with spin-orbit coupling," J. Chem. Theory Comput. **10**, 5567–5576 (2014).

[114]Z. Cao, F. Wang, and M. Yang, "Coupled-cluster method for open-shell heavy-element systems with spin-orbit coupling," J. Chem. Phys. **146**, 134108 (2017).

[115]C. Hampel, K. A. Peterson, and H.-J. Werner, "A comparison of the efficiency and accuracy of the quadratic configuration interaction (QCISD), coupled cluster (CCSD), and Brueckner coupled cluster (BCCD) methods," Chem. Phys. Lett. **190**, 1–12 (1992).

[116]J. Gauss and J. F. Stanton, "Coupled-cluster calculations of nuclear magnetic resonance chemical shifts," J. Chem. Phys. **103**, 3561–3577 (1995).

[117]W. Liu, "Ideas of relativistic quantum chemistry," Mol. Phys. **108**, 1679–1706 (2010).

[118]M. Dolg, "Effective core potentials," in *Modern methods and algorithms of quantum chemistry* (2000) Chap. 3, pp. 507–540.

[119]G. C. Wick, "The Evaluation of the Collision Matrix," Phys. Rev. **80**, 268–272 (1950).

[120]N. Hugenholtz, "Perturbation theory of large quantum systems," Physica **23**, 481–532 (1957).

[121]L. Gyevi-Nagy, M. Kállay, and P. R. Nagy, "Integral-Direct and Parallel Implementation of the CCSD(T) Method: Algorithmic Developments and Large-Scale Applications," J. Chem. Theory Comput. **16**, 366–384 (2020).

[122]H. C. Wong and J. Paldus, "Computer generation of Feynman diagrams for perturbation theory II. Program description," Comput. Phys. Comm. **6**, 9–16 (1973).

[123]C. L. Janssen and H. F. Schaefer, "The automated solution of second quantization equations with applications to the coupled cluster approach," Theor. Chem. Acta **79**, 1–42 (1991).

[124]S. Hirata, "Tensor contraction engine: Abstraction and automated parallel implementation of configuration-interaction, coupled-cluster, and many-body perturbation theories," J. Phys. Chem. A **107**, 9887–9897 (2003).

[125]A. A. Auer, G. Baumgartner, D. E. Bernholdt, A. Bibireata, V. Choppella, D. Cociorva, X. Gao, R. Harrison, S. Krishnamoorthy, S. Krishnan, C. C. Lam, Q. Lu, M. Nooijen, R. Pitzer, J. Ramanujam, P. Sadayappan, and A. Sibiryakov, "Automatic code generation for many-body electronic structure methods: The tensor contraction engine," Mol. Phys. **104**, 211–228 (2006).

[126]D. Datta and J. Gauss, "A non-antisymmetric tensor contraction engine for the automated implementation of spin-adapted coupled cluster approaches," J. Chem. Theory Comput. **9**, 2639–2653 (2013).

[127]J. Zhao and G. E. Scuseria, available at http://jz21.web.rice.edu/drudge/ (Unpublished).

[128]E. Neuscamman, T. Yanai, and G. K. L. Chan, "Quadratic canonical transformation theory and higher order density matrices," J. Chem. Phys. **130** (2009).

[129]L. Huntington, "Development of an Automatic Code Generator and Implementation of Multireference Equation of Motion Coupled-Cluster Theory in the ORCA Program Package, 58," 58–63 (2015).

[130]M. K. MacLeod and T. Shiozaki, "Communication: Automatic code generation enables nuclear gradient computations for fully internally contracted multireference theory," J. Chem. Phys. **142** (2015).

[131]L. K. Sørensen, T. Fleig, and J. Olsen, "A relativistic four-and two-component generalized-active-space coupled cluster method," in *Progress in physical chemistry volume 3* (Oldenbourg Wissenschaftsverlag, 2011), pp. 381–390.

[132]N. C. Rubin and A. E. DePrince III, *P † q: a tool for prototyping many-body methods for quantum chemistry*, 2021.

[133]H. Murayama, "Http://hitoshi.berkeley.edu/129a/dirac.pdf; accessed june 2021,"

[134]M. Born and R. Oppenheimer, "Zur quantentheorie der molekeln," Annalen der Physik **389**, 457–484 (1927).

[135]R. E. Stanton and S. Havriliak, "Kinetic balance: A partial solution to the problem of variational safety in Dirac calculations," J. Chem. Phys. **81**, 1910–1918 (1984).

[136]J. Liu and L. Cheng, "An atomic mean-field spin-orbit approach within exact two-component theory for a non-perturbative treatment of spin-orbit coupling," J. Chem. Phys. **148**, 144108 (2018).

[137]A. Szabo and N. S. Ostlund, *Modern quantum chemistry: introduction to advanced electronic structure theory* (Courier Corporation, 2012).

[138]M. Nooijen*, K. Shamasundar, and D. Mukherjee, "Reflections on size-extensivity, size-consistency and generalized extensivity in many-body theory," Mol. Phys. **103**, 2277–2298 (2005).

[139]A. Asthana, available at https://github.com/aasthan4/autogen/tree/master/tests/ (Unpublished).

[140]I. Shavitt, "The method of configuration interaction," in *Methods of electronic structure theory* (Springer, 1977), pp. 189–275.

[141]M. E. Harding, T. Metzroth, J. Gauss, and A. A. Auer, "Parallel Calculation of CCSD and CCSD(T) Analytic First and Second Derivatives," J. Chem. Theory Comput. **4**, 64–74 (2008).

[142]T. Janowski and P. Pulay, "Efficient parallel implementation of the CCSD external exchange operator and the perturbative triples (T) Energy calculation," J. Chem. Theory Comput. **4**, 1585–1592 (2008).

[143]Z. Pillió, A. Tajti, and P. G. Szalay, "Efficient sparse matrix algorithm to speed up the calculation of the ladder term in coupled cluster programs," J. Chem. Theory Comput. **8**, 3108–3118 (2012).

[144]J. Liu, Y. Shen, A. Asthana, and L. Cheng, "integrals Two-component relativistic coupled-cluster methods using mean-field spin-orbit integrals," J. Chem. Phys. **148**, 034106 (2018).

[145]A. Asthana, J. Liu, and L. Cheng, "Exact two-component equation-of-motion coupled-cluster singles and doubles method using atomic mean-field spin-orbit integrals," J. Chem. Phys. **150**, 1–29 (2019).

[146]T. Saue and L. Visscher, "Four-Component Electronic Structure Methods for Molecules.," in *Kaldor u., wilson s. (eds) theoretical chemistry and physics of heavy and superheavy elements. progress in theoretical chemistry and physics* (Springer, Dordrecht, 2003).

[147]J. L. Whitten, "Coulombic potential energy integrals and approximations," J. Chem. Phys. **58**, 4496–4501 (1973).

[148]B. Dunlap, J. Connolly, and J. Sabin, "On first-row diatomic molecules and local density models," J. Chem. Phys. **71**, 4993–4999 (1979).

[149]K. Eichkorn, O. Treutler, H. Öhm, M. Häser, and R. Ahlrichs, "Auxiliary basis sets to approximate coulomb potentials," Chem. Phys. Lett. **240**, 283–290 (1995).

[150]U. Bozkaya and C. D. Sherrill, "Analytic energy gradients for the coupled-cluster singles and doubles with perturbative triples method with the density-fitting approximation," J. Chem. Phys. **147**, 044104 (2017).

[151]A. E. DePrince III and C. D. Sherrill, "Accuracy and efficiency of coupled-cluster theory using density fitting/cholesky decomposition, frozen natural orbitals, and at 1-transformed hamiltonian," J. Chem. Theory Comput. **9**, 2687–2696 (2013).

[152]F. Aquilante, L. Boman, J. Boström, H. Koch, R. Lindh, A. S. de Merás, and T. B. Pedersen, "Cholesky decomposition techniques in electronic structure theory," in *Linear-scaling techniques in computational chemistry and physics* (Springer, 2011), pp. 301–343.

[153]S. D. Folkestad, E. F. Kjønstad, and H. Koch, "An efficient algorithm for Cholesky decomposition of electron repulsion integrals," J. Chem. Phys. **150** (2019).

[154]N. H. Beebe and J. Linderberg, "Simplifications in the generation and transformation of two-electron integrals in molecular calculations," J. Quantum Chem. **12**, 683–705 (1977).

[155]H. Koch, A. Sánchez de Merás, and T. B. Pedersen, "Reduced scaling in electronic structure calculations using cholesky decompositions," J. Chem. Phys. **118**, 9481–9484 (2003).

[156]X. Feng, E. Epifanovsky, J. Gauss, and A. I. Krylov, "Implementation of analytic gradients for ccsd and eom-ccsd using cholesky decomposition of the electron-repulsion integrals and their derivatives: theory and benchmarks," J. Chem. Phys. **151**, 014110 (2019).

[157]T. Janowski, A. R. Ford, and P. Pulay, "Parallel calculation of coupled cluster singles and doubles wave functions using array files," J. Chem. Theory Comput. **3**, 1368–1377 (2007).

[158]Z. Li, Y. Xiao, and W. Liu, "On the spin separation of algebraic two-component relativistic Hamiltonians: Molecular properties," J. Chem. Phys. **141**, 1–19 (2014).

[159]L. Cheng and J. Gauss, "Perturbative treatment of spin-orbit coupling within spin-free exact two-component theory," J. Chem. Phys. **141** (2014).

[160]Z. Li and W. Liu, "Spin separation of relativistic Hamiltonians," in *Handbook of relativistic quantum chemistry* (Springer, Berlin, Heidelberg, 2015), pp. 1–33.

[161]T. Shen, Z. Zhu, I. Y. Zhang, and M. Scheffler, "Massive-Parallel Implementation of the Resolution-of-Identity Coupled-Cluster Approaches in the Numeric Atom-Centered Orbital Framework for Molecular Systems," J. Chem. Theory Comput. **15**, 4721–4734 (2019).

[162]C. Peng, J. A. Calvin, and E. F. Valeev, "Coupled-cluster singles, doubles and perturbative triples with density fitting approximation for massively parallel heterogeneous platforms," J. Quantum Chem. **119**, 1–13 (2019).

[163]A. P. Rendell, T. J. Lee, and R. Lindh, "Quantum chemistry on parallel computer architectures: coupled-cluster theory applied to the bending potential of fulminic acid," Chem. Phys. Lett. **194**, 84–94 (1992).

[164]T. J. Lee, A. P. Rendell, and P. R. Taylor, "Comparison of the quadratic configuration interaction and coupled-cluster approaches to electron correlation including the effect of triple excitations," J. Phys. Chem. **94**, 5463–5468 (1990).

[165]A. P. Rendell, T. J. Lee, A. Komornicki, and S. Wilson, "Evaluation of the contribution from triply excited intermediates to the fourth-order perturbation theory energy on Intel distributed memory supercomputers," Theor. Chem. Acta **84**, 271–287 (1993).

[166]J. Paldus and H. C. Wong, "Computer generation of Feynman diagrams for perturbation theory I. General algorithm," Comput. Phys. Comm. **6**, 1–7 (1973).

[167]M. Kállay, P. R. Nagy, D. Mester, Z. Rolik, G. Samu, J. Csontos, J. Csóka, P. B. Szabó, L. Gyevi-Nagy, B. Hégely, I. Ladjánszki, L. Szegedy, B. Ladóczki, K. Petrov, M. Farkas, P. D. Mezei, and Á. Ganyecz, "The MRCC program system: Accurate quantum chemistry from water to proteins," J. Chem. Phys. **152** (2020).

[168]J. Olsen, "The initial implementation and applications of a general active space coupled cluster method," J. Chem. Phys. **113**, 7140–7148 (2000).

[169]S. Hirata, T. Yanai, R. J. Harrison, M. Kamiya, and P. D. Fan, "High-order electron-correlation methods with scalar relativistic and spin-orbit corrections," J. Chem. Phys. **126** (2007).

[170]H. S. Nataraj, M. Kllay, and L. Visscher, "General implementation of the relativistic coupled-cluster method," J. Chem. Phys. **133** (2010).

[171]J. Zhao and G. E. Scuseria, available at http://jz21.web.rice.edu/gristmill/ (Unpublished).

[172]Y. Qiu, T. M. Henderson, J. Zhao, and G. E. Scuseria, "Projected coupled cluster theory," J. Chem. Phys. **147** (2017).

[173]R. Schutski, J. Zhao, T. M. Henderson, and G. E. Scuseria, "Tensor-structured coupled cluster theory," J. Chem. Phys. **147**, 184113 (2017).

[174]C. Song, T. Martínez, and N. J., "An automatic differentiation and diagrammatic notation approach for developing analytical gradients of tensor hyper-contracted electronic structure methods," ChemRxiv **107**, 7943–7950 (2021).

[175]A. S. Abbott, B. Z. Abbott, J. M. Turney, and H. F. Schaefer III, "Arbitrary-order derivatives of quantum chemical methods via automatic differentiation," J. Phys. Chem. Lett. **12**, 3232–3239 (2021).

[176]A. Engels-Putzka and M. Hanrath, "A fully simultaneously optimizing genetic approach to the highly excited coupled-cluster factorization problem," J. Chem. Phys. **134** (2011).

[177]T. D. Crawford, T. J. Lee, and H. F. Schaefer, "A new spin-restricted triple excitation correction for coupled cluster theory," J. Chem. Phys. **107**, 7943–7950 (1997).

[178]X. Li and J. Paldus, "Automation of the implementation of spin-adapted open-shell coupled-cluster theories relying on the unitary group formalism," J. Chem. Phys. **101**, 8812–8826 (1994).

[179]J. A. Parkhill and M. Head-Gordon, "A sparse framework for the derivation and implementation of fermion algebra," Mol. Phys. **108**, 513–522 (2010).

[180]W. Kutzelnigg and D. Mukherjee, "Normal order and extended Wick theorem for a multiconfiguration reference wave function," J. Chem. Phys. **107**, 432–449 (1997).

[181]T. Shiozaki, M. Kamiya, S. Hirata, and E. F. Valeev, "Explicitly correlated coupled-cluster singles and doubles method based on complete diagrammatic equations," J. Chem. Phys. **129** (2008).

[182]M. Nooijen and V. Lotrich, "Extended similarity transformed equation-of-motion coupled cluster theory (extended-STEOM-CC): applications to doubly excited states and transition metal compounds," J. Chem. Phys. **113**, 494–507 (2000).

[183]M. Nooijen, "State selective equation of motion coupled cluster theory: Somepreliminaryresults," Int. J. Mol. Sci. **3**, 656–675 (2002).

[184]L. Kong, K. R. Shamasundar, O. Demel, and M. Nooijen, "State specific equation of motion coupled cluster method in general active space," J. Chem. Phys. **130** (2009).

[185]M. Nooijen, O. Demel, D. Datta, L. Kong, K. R. Shamasundar, V. Lotrich, L. M. Huntington, and F. Neese, "Communication: Multireference equation of motion coupled cluster: A transform and diagonalize approach to electronic structure," J. Chem. Phys. **140** (2014).

[186]J. C. Phys, "Multireference equation-of-motion coupled cluster theory," J. Chem. Phys. **204107**, 204107 (2012).

[187]available at http://github.com/aasthan4/autogen (Unpublished).

[188]J. Liu, A. Asthana, L. Cheng, and D. Mukherjee, "Unitary coupled-cluster based self-consistent polarization propagator theory: A third-order formulation and pilot applications," J. Chem. Phys. **148** (2018).

[189]A. G. Taube and R. J. Bartlett, "New perspectives on unitary coupled-cluster theory," Int. J. Quantum Chem. **106**, 3393–3401 (2006).

[190]J. D. Watts, G. W. Trucks, and R. J. Bartlett, "The unitary coupled-cluster approach and moleuclar properties. Applications of the UCC(4) method," Chem. Phys. Lett. **157**, 359–366 (1989).

[191]R. J. Bartlett, S. A. Kucharski, and J. Noga, "Alternative coupled-cluster anstze II. The unitary coupled-cluster method," Chem. Phys. Lett. **155**, 133–140 (1989).

[192]M. R. Hoffmann and J. Simons, "A unitary multiconfigurational coupled-cluster method: theory and applications," J. Chem. Phys. **88**, 993–1002 (1988).

[193]W. Kutzelnigg and S. Koch, "Quantum chemistry in fock space. ii. effective hamiltonians in fock space," J. Chem. Phys. **79**, 4315–4335 (1983).

[194]W. Kutzelnigg, "Error analysis and improvements of coupled-cluster theory," Theoretica chimica acta **80**, 349–386 (1991).

[195]A. Asthana, available at http://github.com/aasthan4/autogen/tree/master/ucc_terms (Unpublished).

[196]W. Meyer, "PNO-CI studies of electron correlation effects. I. Configuration expansion by means of nonorthogonal orbitals, and application to the ground state and ionized states of methane," J. Chem. Phys. **1017**, 834–840 (1973).

[197]C. Riplinger and F. Neese, "An efficient and near linear scaling pair natural orbital based local coupled cluster method," J. Chem. Phys. **138**, 034106 (2013).

[198]C. Riplinger, P. Pinski, U. Becker, E. F. Valeev, and F. Neese, "Sparse maps—a systematic infrastructure for reduced-scaling electronic structure methods. ii. linear scaling domain based pair natural orbital coupled cluster theory," J. Chem. Phys. **144**, 024109 (2016).

[199]C. Peng, J. A. Calvin, F. Pavosevic, J. Zhang, and E. F. Valeev, "Massively parallel implementation of explicitly correlated coupled-cluster singles and doubles using tiledarray framework," J. Phys. Chem. A **120**, 10231–10244 (2016).

[200]A. P. Rendell, T. J. Lee, A. Komornicki, and S. Wilson, "Evaluation of the contribution from triply excited intermediates to the fourth-order perturbation theory energy on intel distributed memory supercomputers," Theor. Chem. Acta **84**, 271–287 (1993).

[201]R. Kobayashi and A. P. Rendell, "A direct coupled cluster algorithm for massively parallel computers," Chem. Phys. Lett. **265**, 1–11 (1997).

[202]L. Gyevi-Nagy, M. Kállay, and P. R. Nagy, "Accurate reduced-cost ccsd (t) energies: parallel implementation, benchmarks, and large-scale applications," J. Chem. Theory Comput. **17**, 860–878 (2021).

[203]V. M. Anisimov, G. H. Bauer, K. Chadalavada, R. M. Olson, J. W. Glenski, W. T. Kramer, E. Apra, and K. Kowalski, "Optimization of the coupled cluster implementation in nwchem on petascale parallel architectures," J. Chem. Theory Comput. **10**, 4307–4316 (2014).

[204]A. Asadchev and M. S. Gordon, "Fast and flexible coupled cluster implementation," J. Chem. Theory Comput. **9**, 3385–3392 (2013).

[205]J. Lee, W. J. Huggins, M. Head-Gordon, and K. B. Whaley, "Generalized Unitary Coupled Cluster Wave functions for Quantum Computation," J. Chem. Theory Comput. **15**, 311–324 (2019).

[206]G. Harsha, T. Shiozaki, and G. E. Scuseria, "On the difference between variational and unitary coupled cluster theories," J. Chem. Phys. **148** (2018).

[207]M. Protocol, C. Cao, Y.-w. Chen, Y. Wu, and E. Deumens, "OPAL : A Multiscale Multicenter Simulation Package Based on MPI-2 Protocol," J. Quantum Chem. **111**, 4020–4029 (2006).

[208]J. S. Kottmann, A. Anand, and A. Aspuru-Guzik, "A feasible approach for automatically differentiable unitary coupled-cluster on quantum computers," Chemical Science **12**, 3497–3508 (2021).

[209]F. A. Evangelista, G. K.-L. Chan, and G. E. Scuseria, "Exact parameterization of fermionic wave functions via unitary coupled cluster theory," J. Chem. Phys. **151**, 244112 (2019).

[210]J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, "Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz," Quantum Science and Technology **4**, 014008 (2018).

[211]H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, "An adaptive variational algorithm for exact molecular simulations on a quantum computer," Nature communications **10**, 1–9 (2019).

[212]H. R. Grimsley, D. Claudino, S. E. Economou, E. Barnes, and N. J. Mayhall, "Is the trotterized uccsd ansatz chemically well-defined?" J. Chem. Theory Comput. **16**, 1–6 (2019).

[213]S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, "Quantum computational chemistry," Rev. Mod. Phys. **92**, 015003 (2020).

[214]S. Thomas, F. Hampe, S. Stopkowicz, and J. Gauss, "Complex ground-state and excitation energies in coupled-cluster theory," arXiv preprint arXiv:2106.03757 (2021).

[215]C. Hättig, "Structure optimizations for excited states with correlated second-order methods: cc2 and adc (2)," Advances in quantum chemistry **50**, 37–60 (2005).

[216]A. Köhn and A. Tajti, "Can coupled-cluster theory treat conical intersections?" J. Chem. Phys. **127**, 044105 (2007).

[217]E. F. Kjønstad, R. H. Myhre, T. J. Martınez, and H. Koch, "Crossing conditions in coupled cluster theory," J. Chem. Phys. **147**, 164105 (2017).

[218]C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, et al., "Array programming with numpy," Nature **585**, 357–362 (2020).

[219]J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," Nat. Comm. **9**, 1–7 (2018).

[220]D. Mukherjee, "Normal ordering and a wick-like reduction theorem for fermions with respect to a multi-determinantal reference state," Chem. Phys. Lett. **274**, 561–566 (1997).

[221]W. Kutzelnigg and D. Mukherjee, "Cumulant expansion of the reduced density matrices," J. Chem. Phys. **110**, 2800–2809 (1999).

[222]W. Kutzelnigg, K. Shamasundar, and D. Mukherjee, "Spinfree formulation of reduced density matrices, density cumulants and generalised normal ordering," Mol. Phys. **108**, 433–451 (2010).

Appendix I

Details about the SO-CCSD(T) working equations

The SO-CCSD equations take the same form as in spin-orbital based CCSD method. The T_1 equations as used in literature is given by

$$\begin{aligned} t_i^a(f_{ii} - f_{aa}) = f_{ia} + \sum_e t_i^e \tilde{f}_{ae} + \sum_m t_m^a \tilde{f}_{me} - \sum_{nf} t_n^f \langle na||if\rangle \\ - \frac{1}{2}\sum_{mef} t_{im}^{ef}\langle ma||ef\rangle - \frac{1}{2}\sum_{men} t_{mn}^{ae}\langle nm||ei\rangle. \end{aligned} \tag{I.1}$$

The T_2 equations can be written as

$$\begin{aligned} t_{ij}^{ab}(f_{ii} + f_{jj} - f_{aa} - f_{bb}) = \langle ij||ab\rangle + P\_(ab)\sum_e t_{ij}^{ae}(\tilde{f}_{be} - \frac{1}{2}\sum_m t_m^b \tilde{f}_{me}) \\ - P\_(ij)\sum_m t_{im}^{ab}(\tilde{f}_{mj} + \frac{1}{2}\sum_e t_j^e \tilde{f}_{me}) + \frac{1}{2}\sum_{mn}\tau_{mn}^{ab} W_{mnij} \\ + \frac{1}{2}\sum_{ef}\tau_{ij}^{ef}\langle ab||ef\rangle + P\_(ij)P\_(ab)\sum_{me}(t_{im}^{ae} W_{mbej} - t_i^e tm^a \langle mb||ej\rangle) \\ + P\_(ij)\sum_e t_i^e \langle ab||ej\rangle - P\_(ab)\sum_m t_m^a(\langle mb||ij\rangle + \frac{1}{2}\sum_{ef}\tau_{ij}^{ef}\langle mb||ef\rangle). \end{aligned} \tag{I.2}$$

The intermediates in the above equations are given by

$$\begin{aligned}\tilde{f}_{ae} &= (1-\delta_{ae})f_{ae} - \frac{1}{2}\sum_{n} f_{me}t_m^a + \sum_{mf} t_m^f\langle ma||fe\rangle \\ &\quad - \frac{1}{2}\sum_{mnf} t\tilde{a}u_{mn}^{af}\langle mn||ef\rangle,\end{aligned} \tag{I.3}$$

$$\begin{aligned}\tilde{f}_{mi} &= (1-\delta_{mi})f_{mi} - \frac{1}{2}\sum_{e} f_{me}t_i^e + \sum_{en} t_n^e\langle mn||ie\rangle \\ &\quad + \frac{1}{2}\sum_{nef} \tilde{\tau}_{in}^{ef}\langle mn||ef\rangle,\end{aligned} \tag{I.4}$$

$$\tilde{f}_{me} = f_{me} + \sum_{nf} t_n^f\langle mn||ef\rangle, \tag{I.5}$$

$$W_{mnij} = \langle mn||ij\rangle + P\_(ij)\sum_{e} t_j^e\langle mn||ie\rangle + \frac{1}{2}\tau_{ij}^{ef}\rangle mn||ef\rangle, \tag{I.6}$$

$$\begin{aligned}W_{mbej} &= \langle mnej\rangle + \sum_{f} t_j^f\langle mb||ef\rangle - \sum_{n} t_n^b\langle mn||ej\rangle \\ &\quad - \sum_{nf}(\frac{1}{2}t_{jn}^{fb} - t^f + jt_n^b)\langle mn||ef\rangle,\end{aligned} \tag{I.7}$$

$$\begin{aligned}\tilde{\tau}_{ij}^{ab} &= t_{ij}^{ab} + \frac{1}{2}(t_i^a t_j^b - t_i^b t_j^a), \\ \tau_{ij}^{ab} &= t_{ij}^{ab} + t_i^a t_j^b - t_i^b t_j^a.\end{aligned} \tag{I.8}$$

$P\_(pq) = 1 - \tilde{p}(pq)$ is the anti-symmetrization operator, where $\tilde{p}(pq)$ interchanges the indices p and q. As in our earlier paper, the intermediate W_{abef} is not used in the above equations. This is because we have used AO-based algorithm for the computation of this intermediate, eliminating the need for its storage.
Once the amplitudes t_i^a and t_{ij}^{ab} are determined through above equations, the CCSD energy is calculated as

$$E_{CCSD} = \sum_{ia} t_i^a f_{ia} + \frac{1}{4}\sum_{ijab} \tau_{ij}^{ab}\langle ij||ab\rangle. \tag{I.9}$$

The $E_{(T)}$ correction is given by the equations

$$E_{(T)} = \frac{1}{36}\sum_{ijkabc} t(c)_{ijk}^{abc} D_{ijk}^{abc}(t(c)_{ijk}^{abc} + t(d)_{ijk}^{abc}). \tag{I.10}$$

Here $t(c)_{ijk}^{abc}$ is the connected term, which arises from leading fourth order correction to energy from triple excitation. It is given by

$$t(c)_{ijk}^{abc} = \frac{1}{D_{ijk}^{abc}}(\sum_{e} P(i/jk)P(a/bc)t_{jk}^{ae}\langle bc||ei\rangle + \tag{I.11}$$

$$P(i/jk)P(a/bc)t_{mi}^{bc}\langle jk||ma\rangle), \tag{I.12}$$

where,

$$D_{ijk}^{abc} = f_{ii} + f_{jj} + f_{kk} - f_{aa} - f_{bb} - f_{cc}. \tag{I.13}$$

$t(d)_{ijk}^{abc}$ is the disconnected term, which arises from one of the leading order fifth order correction to energy from triple excitation. It is given by

$$t(d)_{ijk}^{abc} = \frac{P(i/jk)P(a/bc)t_i^a\langle bc||jk\rangle}{D_{ijk}^{abc}}. \tag{I.14}$$

Here $P(p/qr) = 1 - \tilde{p}(pq) - \tilde{p}(pr)$.

Appendix II

Details about the SO-EOM-CCSD working equations

The SO-EOM-CCSD working equations take the same form as in the case of spin-orbital based non-relativistic EOM-CCSD method. The residual vectors for EOM-CCSD secular equations can be written as [116]

$$\begin{aligned}\tilde{r}_i^a =& \langle \Psi_i^a | (\bar{H} - E) R | \Psi_{HF} \rangle \\ =& - E_{ex} r_i^a + \sum_c \bar{H}_{a,c} r_i^c - \sum_k \bar{H}_{k,i} r_k^a + \sum_{kc} \bar{H}_{k,c} r_{ik}^{ac} \\ &+ \sum_{kc} \bar{H}_{ak,ic} r_k^c + \frac{1}{2} \sum_{kcd} \bar{H}_{ak,cd} r_{ik}^{cd} - \frac{1}{2} \sum_{kld} \bar{H}_{kl,ed} r_{kl}^{ad},\end{aligned} \tag{II.1}$$

and

$$\begin{aligned}\tilde{r}_{ij}^{ab} =& \langle \Psi_{ij}^{ab} | (\bar{H} - E) R | \Psi \rangle \\ =& - E_{ex} r_{ij}^{ab} + P_(ab) \sum_e \bar{H}_{b,e} r_{ij}^{ae} - P(ij) \sum_k \bar{H}_{k,j} r_{ik}^{ab} \\ &+ \frac{1}{2} \sum_{kl} \bar{H}_{kl,ij} r_{kl}^{ab} + \frac{1}{2} \sum_{ef} \bar{H}_{ab,ef} r_{ij}^{ef} + P(ij) P(ab) \sum_{kc} \bar{H}_{bk,jc} r_{ik}^{ac} \\ &- P(ij) \sum_c \bar{H}_{ab,cj} r_j^c + P(ab) \sum_k \bar{H}_{ak,ij} r_k^b \\ &+ \sum_{kc} \bar{H}_{abk,ijk} r_k^c + P(ij) \frac{1}{2} \sum_{kcd} \bar{H}_{abk,icd} r_{jk}^{cd} r_{jk}^{cd} - P(ab) \frac{1}{2} \sum_{kld} \bar{H}_{akl,ijd} r_{kl}^{bd},\end{aligned} \tag{II.2}$$

where E_{ex} represents the excitation energy. Apart from intermediates already defined in Eq. I.4 to I.8, the following intermediates are used

$$\begin{aligned}\bar{H}_{ab,cj} =&\langle ab||cj\rangle - \sum_m \bar{H}_{mc} t^{ab}_{mj} + \sum_f t^f_j \bar{H}_{ab,cf} + \frac{1}{2}\sum_{mn}\langle mn||cj\rangle \tau^{ab}_{mn} \\ &- P(ab)\sum_{mf}\langle mb||cf\rangle t^{af}_{mj} - P(ab)\sum_m t^a_m\{\langle mb||cj\rangle - \sum_{nf} t^{bf}_{nj}\langle mn||cf\rangle\}.\end{aligned} \tag{II.3}$$

$$\bar{H}_{ab,ef} =\langle ab||ef\rangle - P(ab)\sum_m t^b_m\langle am||ef\rangle + \frac{1}{2}\tau^{ab}_{mn}\langle mn||ef\rangle, \tag{II.4}$$

$$\bar{H}_{mn,ie} =\langle mn||ie\rangle - \sum_f t^f_i\langle mn||fe\rangle, \tag{II.5}$$

$$\bar{H}_{am,ef} =\langle am||ef\rangle - \sum_n t^a_n\langle nm||ef\rangle, \tag{II.6}$$

$$\bar{H}_{mb,ij} =\langle mb||ij\rangle - \sum_e f_{me} t^{be}_{ij}, \tag{II.7}$$

while the three body intermediates are defined as

$$\bar{H}_{abk,ijc} =p(ab)\sum_d \bar{H}_{kb,cd} t^{ad,ij} - P(ij)\sum_l \bar{H}_{kl,cj} t^{ab}_{il}, \tag{II.8}$$

$$\bar{H}_{abk,icd} = -\, sum_l \langle kl||dc\rangle t^{ab}_{il}, \tag{II.9}$$

$$\bar{H}_{akl,ijd} = \sum_e \langle kl||ed\rangle t^{ae}_{ij}. \tag{II.10}$$

Only two terms in the amplitude equations involve the four-particle matrices $\langle ab||cd\rangle$ and $\bar{H}_{ab,cd}$. The first of these terms is the fifth contraction in Eq. II.2, namely $\frac{1}{2}\sum_{ef}\bar{H}_{ab,ef} r^{ef}_{ij}$. The second term is the seventh contraction on the right side of Eq. II.2, namely $P(ij)\sum_c \bar{H}_{ab,cj} r^c_i$.

Curriculum Vitae **Ayush Asthana** **June, 2021**

505 West University Parkway, A22,
Baltimore, Maryland 21210 USA
(+1) 443-708-6400
ayushasthana15@gmail.com

EDUCATION AND DEGREES

2016–Present Graduate student, Department of Chemistry
Johns Hopkins University

2011–2016 BS-MS Dual Degree (Chemistry), Department of Chemistry
Indian Institute of Technology Kanpur

RESEARCH EXPERIENCE

Johns Hopkins University Department of Chemistry *(2016-Present)*
Graduate Student in the laboratory of a Prof. Lan Cheng

Indian Association for Cultivation of Sciences *(January 2016-May 2016)*
Post-baccalaureate Researcher in the laboratory of a Prof. Debashis Mukherjee

Virginia Tech *(May 2014-July 2014)*
Summer research fellow in the laboratory of a Prof. T. Daniel Crawford

TEACHING EXPERIENCE

Johns Hopkins University, *(August 2016-April 2018)*
Teaching Assistant

- Graduate Student Instructor for classes Introduction to physical chemistry lab, Introduction to organic chemistry lab, Introduction to computational chemistry.

PUBLICATIONS

- J. Liu, X. Zheng, **A. Asthana**, C. Zhang, and L. Cheng, "Analytic Evaluation of Energy First Derivatives for Spin-Orbit Coupled-Cluster Singles and Doubles Augmented with Noniterative Triples Method: General Formulation and An Implementation for First-Order Properties", J. Chem. Phys. **154**, 064110 (2021).
- G. Liu, C. Zhang, S. Ciborowski, **A. Asthana**, L. Cheng and K. Bowen, "Mapping the Electronic Structure of the Uranium (VI) Dinitride Molecule, UN2", J. Phys. Chem. A, **124**, 6486 (2020).
- **A. Asthana**, J. Liu, and L. Cheng, "Exact two-component equation-of-motion coupled-cluster singles and doubles method using atomic mean-field spin-orbit integrals", J. Chem. Phys. **150**, 074102 (2019).
- J. Liu, **A. Asthana**, L. Cheng, and D. Mukherjee, "Unitary coupled-cluster based self-consistent polarization propagator theory: A third-order formulation and pilot applications",J. Chem. Phys. **148**, 244110 (2018).
- J. Liu, Y. Shen, **A. Asthana**, and L. Cheng, "Two-component relativistic coupled-cluster methods using mean-field spin-orbit integrals", J. Chem. Phys. **148**, 034106 (2018).

PRESENTATIONS

- Relativistic coupled-cluster methods for heavy-element computational chemistry (Invited talk). IBM Almaden Research Center (Invited), Jul 2020
- New algorithmic development for relativistic equation-of-motion coupled-cluster method (Poster presentation). American Chemical Society National Meeting and Exposition, San Francisco, CA, Aug 2020
- Development of relativistic quantum chemistry methods for molecules containing heavy-elements (Graduate Board Oral talk). Johns Hopkins University, Dec 2019

- Chemistry at ultracold temperatures (Department seminar). Johns Hopkins University, Mar 2019
- Extended wick's theorem, spin free cumulants and their role in formulation and analyzing spacial and spin correlation of many electron systems (Master's thesis talk). Indian Institute of Technology Kanpur, Apr 2016